

A Project on

Hostel Management System

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE
BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
SUBMITTED BY

RACHANA PATIL
SADAF SHAIKH
ADITYA SAPKAL
SUJAL THAKRE

EXAM NO: B191104252
EXAM NO: B191104263
EXAM NO: B191104261
EXAM NO: B191104270

Under the guidance of

Prof. Pranjali More

Department of Computer Engineering

A.B.M.S.PARISHAD'S

ANANTRAO PAWAR COLLEGE OF
ENGINEERING & RESEARCH

Parvati, Pune-09

NAAC Accredited Institute



SAVITRIBAI PHULE PUNE UNIVERSITY

2023-24

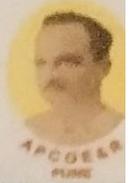


**AKHIL BHARATIYA MARATHA SHIKSHAN PARISHAD'S
ANANTRAO PAWAR COLLEGE OF ENGINEERING & RESEARCH**

Sr. No. 103, Parvati, Pune - 411 009.

Tel.: 020-24218901/8959 Tele Fax : 020-24213929

Web.: <http://www.abmspcoerpune.org> Email : abmspcoe@yahoo.com



Approved by AICTE & Govt. of Maharashtra, Affiliated to Savitribai Phule Pune University

NAAC ACCREDITED, DTE CODE :- EN 6794, AISHE CODE :- C-41484

Savitribai Phule Pune University Identification No. PU/PN/Engg. / 441/2012.

Department of Computer Engineering

CERTIFICATE

This is to certify that the project report entitles

“HOSTEL MANAGEMENT SYSTEM”

Submitted by

RACHANA PATIL

Exam No: B191104252

SADAF SHAIKH

Exam No: B191104263

ADITYA SAPKAL

Exam No: B191104261

SUJAL THAKRE

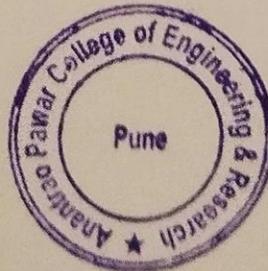
Exam No: B191104270

are bonafide students of this institute and the work has been carried out by them under the supervision of Prof. Pranjali More and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Computer Engineering.

Prof. Pranjali More
Assistant Professor
Computer Engineering Dept.

Prof. Rama Gaikwad
Head of Department
Computer Engineering Dept.

Name and Signature of External Examiner



(Dr. S. B. Thakare)
Principal

Place: Pune

Date: 28/05/24

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of the "Hostel Management System" project. This endeavor was made possible through the combined efforts, dedication, and support of many individuals and organizations. First and foremost, we extend our gratitude to our project guide **Prof. Pranjali More** for their valuable guidance, expertise and unwavering encouragement throughout the project. Their mentorship was instrumental in shaping the project's direction and ensuring its successful execution.

We are deeply thankful to our team members who demonstrated exceptional teamwork and the HOD of computer engineering department **Prof. Rama Gaikwad**, all teaching and non-teaching faculty of APCOER for their commitment, working tirelessly to bring this project to fruition. Their collaboration and dedication were key to overcoming various challenges and achieving the project's objectives.

We also extend our appreciation to **Dr. Sunil B. Thakre**, Principal, Anantrao Pawar college of engineering and Research for providing us with the necessary resources, infrastructure, and access to research materials.

The institution's support was crucial in facilitating our research and development efforts. Our sincere thanks go to the participants who volunteered to be a part of our data collection and testing processes. Their contributions also essential for the accuracy and reliability of our project "Hostel Management System". Furthermore, we want to acknowledge the invaluable insights and inspiration we gained from prior research in this field. Lastly, we extend our gratitude to our friends and family for their patience and understanding throughout this project. Your unwavering support and encouragement motivated us to persevere through the project's various stages. We are truly thankful for all contributions and hope that our "Hostel Management System" project will have a positive impact in the field of facial recognition and security.

Sujal Suresh Thakre

Sadaf Azam Shaikh

Rachana Rajendra Patil

Aditya Krishna Sapkal

ABSTRACT

This project is based on the Design and Implementation of a Hostel Management System for our college hostel. It evolved from the manual file record system used to store hostel details, room details, student records, hostel allocation and other data in the hostel. With the help of the internet, computers and other mobile devices are used to manage hostel records and help the administrator effectively access these records. The goal of this project is to develop a dependable and effective hostel administration system that can easily handle all manual labor. To accomplish its goals, an incremental and iterative design process was used, along with the creation of a database. This project is carried out using Java, Ajax, Python with other front-end and backend technologies using Visual Studio Code as the Integrated Development Environment. Record officers and hostel managers may better monitor and oversee their separate activities with the use of the developed Hostel Management System. Additionally, it is a quicker and more effective approach to maintain and keep an eye on hostel data.

The hostel management system is a comprehensive software solution designed to automate and streamline the administrative tasks involved in managing a hostel. It features automated room allocation, online booking and reservations, integrated financial management, attendance management, and complaint tracking. Additionally, it incorporates security measures. With robust reporting and analytics tools, the system enhances operational efficiency, improves resident satisfaction, and ensures a secure and comfortable living environment. This technology-driven approach minimizes manual efforts, reduces errors, and supports effective decision-making for hostel administrators.

INDEX

1	Synopsis	1
	1.1 Project Title	2
	1.2 Project Option	2
	1.3 Internal Guide	2
	1.4 Technical Keywords	3
	1.5 Problem Statement	3
	1.6 Abstract	3
	1.7 Goals and Objectives	4
	1.8 Relevant mathematics associated with the proj.	4
	1.9 Names of Conferences / Journals where papers can be published	6
	1.10 Review of Conference/Journal Papers supporting Project idea.	7
	1.11 Review of Conference/Journal Papers supporting Project idea.	9
	1.12 Plan of Project Execution	11
2	Technical Keywords	13
	2.1 Area of Project	14
	2.2 Technical Keywords	14
3	Introduction	15
	3.1 Project Idea	16
	3.2 Motivation of the Project	17
	3.3 Literature Survey	17
4	Problem Statement and Scope	23
	4.1 Problem Statement	26
	4.1.1 Goals and Objectives	26
	4.1.2 Statement of Scope	26
	4.2 Major Constraint	28

4.3 Methodologies of Problem Solving and Efficiency Issues	30
4.4 Outcome	38
4.5 Applications	40
4.6 Hardware Resources Required	41
4.7 Software Resource Required	42
5 Project Plan	43
5.1 Project Estimates	44
5.1.1 Reconciled Estimates	45
5.1.2 Project Resources	48
5.2 Risk Management over NP Hard Analysis	49
5.3 Project Schedule	52
5.3.1 Project Task Set	52
5.3.2 Task Network	54
5.4 Team Organization	56
5.4.1 Team Structure	58
5.4.2 Management Reporting and Communication	59
6 Software Requirement Specification	61
6.1 Introduction	62
6.1.1 Purpose and Scope of Document	62
6.1.2 Overview of responsibilities of Developer	63
6.2 Usage Scenario	64
6.2.1 User Profile	65
6.2.2 Proposed System Architecture	66
6.3 Module	67
6.3.1 Data Flow Diagram	67
6.3.2 UML Diagram	67
6.4 Design Constraint	76
6.5 Software Interface Constraint	78

7	Detailed Design Document Using Appendix A and B82
	7.1 Architectural Design83
	7.2 Data Design86
	7.2.1 Internal Software Data Structure87
	7.3 Component Design88
8	Project Implementation91
	8.1 Introduction92
	8.2 Tools and Technologies Used92
	8.3 Methodologies / Algorithms used93
	8.4 Algorithm95
9	Software Testing97
	9.1 Type of Testing Used98
	9.2 Test Cases and Results98
10	Results100
	10.1 Screenshots101
11	Deployment and Maintenance104
	11.1 Installation and Uninstallation105
	11.2 User Help106
12	Conclusion and Future Scope108
	Annexure A References111
	Annexure B Competition Participation Certificates114
	Annexure C Paper, Certificate, Reviewers Comments of Paper Submitted118
	Annexure D Plagiarism Report143
	Annexure E Information of Project Group Members146

Annexure F Final Project Presentation Handouts149

Annexure G Project Achievements167

Annexure H Documentation on7Step-by-Step Execution of171

a Project

Annexure I System Code174

List of Figures

5.1 Task Network60
6.1 System Diagram66
6.2 Data Flow Diagram67
6.3 Activity Diagram69
6.4 Sequence Diagram70
6.5 Use case Diagram73
6.6 Class Diagram75
7.1 Architectural Design83
7.2 System Architectural Design83
10.1 Home Page101
10.2 User Login Page101
10.3 Camera Integration102
10.4 Room Booking102
10.5 Student Attendance Details103
10.6 Total Rooms103
10.7 Review103

List of Tables

1.1 Review of Conference 17
1.2 Review of Conference 29
4.1 Hardware Requirements41
4.2 Software Requirements42

List of Abbreviations

Abbreviations

Full Form

CPU	Central Processing Unit
GPU	Graphics Processing Unit
IDE	Integrated Development Unit
API	Application Programming Interface
GUI	Graphical User Interface
OpenCV	Open-Source Computer Vision library
IP	Internet Protocol
AI	Artificial Intelligence
DOA	Delegation Of Authority
CEO	Chief Executive Officer
API	Application Programming Interface

CHAPTER 1
SYNOPSIS

1.1 PROJECT TITLE

Hostel Management System

1.2 PROJECT OPTION

Basic Attendance Tracking System:

- Develop a simple hostel management system focused primarily on attendance tracking using basic technologies such as RFID or barcode scanners.
- Ideal for small-scale hostel facilities or as a starting point for learning about hostel management systems.

Camera-Based Attendance System:

- Implement a more advanced hostel management system utilizing camera attendance technology for automated attendance tracking.
- Incorporate facial recognition algorithms for accurate identification and recording of student attendance.
- Suitable for mid to large-scale hostel facilities looking to enhance security and efficiency.

Comprehensive Hostel Management System:

- Develop a fully-featured hostel management system covering various aspects such as room allocation, inventory management, billing, and reporting.
- Include modules for student registration, room assignment, maintenance tracking, and staff management.
- Targeted towards larger hostel facilities or institutions with complex management needs.

Mobile Application Integration:

- Create a mobile application interface for the hostel management system, allowing students to check-in/out, view room assignments, and receive notifications.
- Enhance accessibility and convenience for students and staff, enabling them to access hostel-related information from their smartphones.
- Suitable for modernizing hostel management processes and improving user experience.

1.3 INTERNAL GUIDE

Prof. Pranjali More

1.4 TECHNICAL KEYWORDS

Allocation: The term "allocation" refers to the process of distributing resources, tasks, or responsibilities among various parties or uses. It involves assigning specific amounts or portions of something, such as money, time, materials, or labor, to different people, projects, or categories based on certain criteria or priorities.

Database: A database is an organized collection of structured information or data, typically stored electronically in a computer system.

File Management: File management refers to the process of organizing, storing, naming, and handling files on a computer or a digital storage system. Effective file management ensures that files are easily accessible, secure, and properly maintained.

1.5 PROBLELM STATEMENT

Design and develop a Hostel Management System to address the challenges faced by a large hostel facility in efficiently managing its operations. The hostel currently faces several manual and time-consuming processes, leading to inefficiencies, data inaccuracies, and reduced guest satisfaction. The objective is to create a comprehensive software solution that automates key tasks and enhances overall hostel management.

1.6 ABSTRACT

This project is based on the Design and Implementation of an Online Hotel Management System for use at the Mountain Top University Hostel. It evolved from the manual file record system used to store hostel details, room details, student records, hostel allocation and other data in the hostel. With the help of the internet, computers and other mobile devices are used to manage hostel records and help the administrator effectively access these records. The goal of this project is to develop a dependable and effective hostel administration system that can easily handle all manual labor. To accomplish its goals, an incremental and iterative design process was used, along with the creation of a database. This project is carried out using PHP, MySQL with other frontend and backend technologies using Visual Studio Community 2017 as the Integrated Development Environment. Record officers and hostel managers may better monitor and oversee their separate activities with the use of the developed Hostel Management System. Additionally, it is a quicker and more effective approach to maintain and keep an eye on hostel data.

1.7 GOALS AND OBJECTIVES

- **Enhance Transparency:** Implement blockchain technology to provide a transparent and auditable ledger of crowdfunding transactions, ensuring trust and accountability among participants.
- **Reduce Costs:** Minimize fees associated with traditional crowdfunding platforms by eliminating intermediaries and automating processes through smart contracts, thereby increasing the funds available for projects.
- **Expand Access:** Facilitate global participation in crowdfunding campaigns by removing geographical barriers and enabling contributions in various cryptocurrencies, fostering inclusivity and diversity among backers.
- **Mitigate Fraud:** Utilize blockchain's immutability and security features to reduce the risk of fraudulent activities such as misappropriation of funds or manipulation of campaign data, safeguarding the interests of both backers and project creators.
- **Improve Efficiency:** Streamline crowdfunding processes through smart contracts, automating tasks such as fund disbursement, milestone tracking, and voting mechanisms, enhancing efficiency and reducing administrative overhead.
- **Ensure Compliance:** Navigate regulatory complexities and ensure compliance with relevant laws and regulations governing crowdfunding activities in different jurisdictions, fostering a legal framework conducive to decentralized fundraising.
- **Empower Innovation:** Empower entrepreneurs, startups, and creators to access capital and bring their ideas to life by providing a decentralized and inclusive platform for fundraising, fostering innovation and economic growth.

1.8 RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT

1. Database Management

- **Relational Algebra:**
- **Set Theory:** Used for operations like UNION, INTERSECT, and DIFFERENCE on tables.
- **Cartesian Product:** Combining tuples from two relations.
- **Selection and Projection:** Filtering rows and columns in a database.

2. Algorithm Development

- Graph Theory:
 - Graph Representation: Rooms, students, and pathways can be represented as graphs for optimized allocation and navigation.
 - Search Algorithms: Breadth-First Search (BFS) and Depth-First Search (DFS) for finding available rooms or resources.
- Sorting and Searching Algorithms:
 - Binary Search: Efficiently finding a student record or room.
 - Sorting Algorithms: QuickSort, MergeSort for ordering data (e.g., by fee due dates or room numbers).

3. Data Analysis

- Descriptive Statistics:
 - Mean, Median, Mode: For analyzing student data like age, room occupancy rates, or fee payments.
 - Standard Deviation and Variance: Understanding the dispersion in data such as fee payment patterns or room occupancy.
- Inferential Statistics:
 - Hypothesis Testing: Determining if changes (like a new fee structure) have statistically significant effects on payments.
 - Regression Analysis: Predicting trends like future occupancy rates based on historical data.

4. Financial Calculations

- Basic Arithmetic and Algebra:
 - Addition, Subtraction, Multiplication, Division: For fee calculations, total payments, and outstanding balances.
 - Percentage Calculations: Discounts, late fee percentages, and interest rates.

5. Optimization Techniques

- Linear Programming:
 - Objective Function: Minimize costs or maximize room occupancy.

- Constraints: Budget limits, room capacities, maintenance schedules.

6. Data Visualization

- Coordinate Geometry:
 - Graphs and Charts: Plotting occupancy trends, fee collection over time, and maintenance requests.
- Matrix Algebra:
 - Transformation and Rotation: For graphical representations and room layout planning.
- Practical Application in Project

1.9 NAME OF CONFERENCES

- Paper ID: 10

Conference name: National Conference on Futuristic Trends in Engineering & Applied Science 2024 (NCFTEAS 2024)

Paper Title: Hostel Management System

1.10 REVIEW OF CONFERENCE/JOURNAL PAPERS SUPPORTING PROJECT AREA

Sr.no	Paper Title	Author	year	Key findings	limitations
1	Design and Implementation Of Hostel Management System using Java and Mysql	Pooja Nagpal, Nishant Goel, Shivam Sangwan, Himanshi Dixit	2020	Better resource management: Hostel management systems aid in the effective management of resources such as food, utilities, and other amenities, leading to cost savings and improved service delivery.	Data Security Concerns: Storing sensitive information in the system can pose data security risks, necessitating robust security measures to prevent breaches and leaks.
	<p><u>Description:</u> Reservation Management: Facilitates the efficient handling of room reservations, cancellations, and modifications, ensuring a smooth booking process for guests</p>				
2	Hostel Management System and Aggregation	Prof. Shyamsundar Magar Mr. Rohit Jadhav Ms. Sakshi Said Mr. Shashikant Jadhav	2021	Efficient room allocation: Hostel management systems enable optimized room allocation, ensuring that available space is utilized effectively	Initial Cost: Implementing a hostel management system can involve significant upfront costs for software, hardware, and training
	<p><u>Description:</u> Room Allocation: Enables the automated allocation of rooms based on specific criteria such as room type, occupancy, and guest preferences, enhancing the overall guest experience</p>				

3.	Hostel management System	Mr. Kathiresh Ms. Nandika Mr. Naveenkumar Mr. Manikandan	2023	Streamlined communication: These systems facilitate seamless communication between hostel staff, students, and administrators, enhancing overall coordination and reducing miscommunication	Technical Issues: System glitches, software bugs, or hardware malfunctions can disrupt operations and cause inconvenience to both staff and residents
<p>Description: Check-In and Check-Out Processes: Simplifies and accelerates the check-in and check-out procedures for guests, minimizing waiting times and enhancing customer satisfaction</p>					
4.	Online Hostel Management	Abhishek Askar, Saumya Gaur, Sejal Deollikar And Mrs. Archana Ubale	2022	Enhanced parent-student communication: Through features such as online portals and regular updates, hostel management systems facilitate better communication between parents and hostel authorities, providing peace of mind for both parties	Dependency on Technology: Reliance on technology makes the system vulnerable to potential outages or cybersecurity threats, leading to downtime and data breaches.
<p>Description: Billing and Invoicing: * Automates the billing process, generating accurate invoices for guests, and facilitating seamless payment transactions, including various payment methods</p>					
5.	Hostel Management with Integrated System	Dr. J. Sree Rambabu Mr. M. Mohammed Riyaz Mr. D. Rajkumar Mr. Tamilarasan	2023	Real-time monitoring: With the capability for real-time monitoring of hostel activities and operations, these systems enable prompt intervention in case of any issues or emergencies	Training Requirements: Staff members may require extensive training to effectively use the management system, which can be time-consuming and costly
<p>Description: Inventory Management: Tracks and manages inventory levels of essential hostel resources such as bedding, toiletries, and other amenities, ensuring efficient supply chain management</p>					
Table 1.1: Review of Conference 1					

- Paper ID: CST 1111

Conference name: International Conference on Advanced Technologies and Intelligent Automation (ICATIA-24).

Paper Title: Hostel management system

1.11 REVIEW OF CONFERENCE/JOURNAL PAPERS SUPPORTING PROJECT AREA

1.	Hostel Management with Integrated System	Dr. J. Sree Rambabu Mr. M. Mohammed Riyaz Mr. D. Rajkumar Mr. Tamilarasan	2023	Real-time monitoring: With the capability for real-time monitoring of hostel activities and operations, these systems enable prompt intervention in case of any issues or emergencies	Training Requirements: Staff members may require extensive training to effectively use the management system, which can be time-consuming and costly
Description: Inventory Management: Tracks and manages inventory levels of essential hostel resources such as bedding, toiletries, and other amenities, ensuring efficient supply chain management					
2.	Study of Digitalized Hostel Management System	Kartik haudhri, Riddhi Kevat	2021	Enhanced parent-student communication: Through features such as online portals and regular updates, hostel management systems facilitate better communication between parents and hostel authorities, providing peace of	Lack of Customization: Some systems may lack the flexibility to accommodate specific hostel requirements, leading to limitations in customization.
Description: Communication Tools: Facilitates effective communication between hostel management and guests through automated notifications, reminders, and updates regarding bookings and services					
3.	Hostel management System based on Finger Print Authentication	G. Rajkumar and T. Sivagama Sundari	2018	Data-driven insights: By generating detailed reports and analytics, hostel management systems provide valuable insights for	Integration Challenges: Integrating the management system with other existing software

				decision-making, enabling administrators to identify trends and make informed strategic choices	or platforms can be complex and may require additional technical support.
Description: Integration Capabilities: Offers the flexibility to integrate with other relevant systems such as accounting software, customer relationship management (CRM) tools, and online booking platforms, enhancing the overall efficiency and functionality of the management system					
4.	Hostel Management System for Sanskriti School of Engineering	P. Meghana 1 D. Jyothi 2 E. Jahanvi 2 E. Jeevana 2 J. Sunil 2	2018	Increased student satisfaction: By streamlining various processes and enhancing overall efficiency, hostel management systems contribute to a positive student experience, leading to higher satisfaction levels among residents	Maintenance Needs: Regular maintenance and updates are necessary to ensure the smooth functioning of the system, which can add to operational costs
Description: Reporting and Analytics: Generates detailed reports and analytics on various aspects of hostel operations, providing valuable insights for informed decision-making and performance evaluation					

Table 1.2: Review of Conference 2

1.12 PLAN OF PROJECT EXECUTION

1. Initial Planning and Requirement Gathering

Define Objectives: Identify the primary goals of the hostel management system (e.g., streamline hostel operations, improve record-keeping, enhance communication).

Stakeholder Meetings: Meet with stakeholders (hostel administrators, staff, students) to gather requirements and understand their needs.

Scope Definition: Clearly define the scope of the project, including the key features and functionalities.

2. System Design and Architecture

Design Specifications: Create detailed design documents outlining the system architecture, data flow, and database schema.

Wireframes and Mockups: Develop wireframes and mockups for the user interface (UI) to visualize the layout and user experience (UX).

Technology Stack: Choose the technology stack (e.g., programming languages, frameworks, databases) based on project requirements.

3. Database Design

Schema Design: Design the database schema, including tables, relationships, and constraints to ensure data integrity and efficient access.

Data Models: Create data models to represent different entities (e.g., students, rooms, staff, payments) and their relationships.

4. Development Phase

Backend Development:

- Implement the server-side logic, database interactions, and APIs.
- Ensure secure data handling and authentication mechanisms.

Frontend Development:

- Integrate frontend and backend components.
- Implement features such as user registration, room allocation, payment processing, and reporting.

5. Testing Phase

Unit Testing: Test individual components for functionality and performance.

Integration Testing: Ensure different modules work together seamlessly.

User Acceptance Testing (UAT): Conduct testing sessions with actual users to gather feedback and identify issues.

Bug Fixing: Resolve identified bugs and retest to ensure stability.

6. Deployment Phase

Server Setup: Prepare the production environment, including servers, databases, and necessary configurations.

Deployment: Deploy the application to the production environment.

Data Migration: Migrate existing data (if any) to the new system.

Configuration: Configure the system for real-world use, including setting up user accounts and permissions.

7. Training and Documentation

User Training: Conduct training sessions for hostel staff and other users to familiarize them with the system.

Documentation: Provide comprehensive user manuals, technical documentation, and FAQs to assist users and administrators.

CHAPTER 2
TECHNICAL KEYWORDS

2.1 AREA OF PROJECT

The area of the project encompasses several domains, including:

- **Educational Institutions:**

Universities and Colleges: Managing student hostels, dormitories, and residential halls for undergraduate, postgraduate, and doctoral students.

Schools: Administering boarding facilities for students, including primary and secondary schools.

- **Corporate Sector:**

Employee Accommodation: Managing housing for employees, especially in organizations that provide on-site living arrangements for staff, such as oil and gas companies, construction firms, and remote work sites.

Intern and Trainee Housing: Facilitating accommodation for interns, trainees, and employees on temporary assignments.

- **Healthcare Sector:**

Hospital Staff Housing: Managing accommodation for medical staff, including doctors, nurses, and other healthcare workers, especially in large hospitals and medical campuses.

Patient and Family Accommodation: Providing temporary housing for patients and their families, especially in hospitals located far from the patient's home.

2.2 TECHNICAL KEYWORDS

Allocation: The term "allocation" refers to the process of distributing resources, tasks, or responsibilities among various parties or uses. It involves assigning specific amounts or portions of something, such as money, time, materials, or labor, to different people, projects, or categories based on certain criteria or priorities.

Database: A database is an organized collection of structured information or data, typically stored electronically in a computer system.

File Management: File management refers to the process of organizing, storing, naming, and handling files on a computer or a digital storage system.

CHAPTER 3
INTRODUCTION

3.1 PROJECT IDEA

Current system: Does our college already have a hostel management system in place? If so, is it outdated, inefficient, or lacking in features?

Scale of the hostel: For a small hostel with a handful of students, a simpler manual system might suffice. But for a larger college hostel with hundreds of residents, a dedicated system can save significant time and resources.

Integration needs: Does your college have other existing systems, like student portals or finance software? A new hostel management system would ideally integrate with these for smoother data flow.

Here are some alternatives to consider if building a system from scratch isn't the best fit:

Existing Hostel Management Software: There are many commercially available hostel management software options. These might be a quicker and more cost-effective solution, especially for smaller colleges.

Open-source options: Open-source hostel management systems exist that can be customized to fit your college's needs. However, this might require some technical expertise for setup and maintenance. Building your own system can be a great learning experience, but it's important to weigh the need, resources, and available alternatives before diving in.

Here are some situations where building a hostel management system might be a good idea for your college:

Outdated System: If your college's current system is clunky and inefficient, building a new one can significantly improve operations.

Specific Needs: If existing systems lack features important to your college (like guest management or a community forum), building your own allows for customization.

3.2 MOTIVATION OF THE PROJECT

- **Learning in Demand Skills:** Building this system will help you develop practical skills in software development, database management, and potentially even web design if you create a user interface. These skills are highly sought after in today's tech job market.
- **Showcase Your Abilities:** The completed project serves as a tangible portfolio piece demonstrating your technical skills and problem-solving abilities. This can be a major advantage when applying for internships or jobs related to software development.
- **Personalized Solution:** Do you find the current hostel booking system frustrating? This project allows you to design a system that addresses those pain points, creating a solution you'd love to use yourself.
- **Creative Freedom:** While the core functionalities might be similar, you can customize the system with unique features. Maybe it integrates with social media for easier guest interaction, or offers a gamified loyalty program.
- **Entrepreneurial Spirit:** This project can be a springboard for a future business idea. You could develop a commercial hostel management system or offer customization services for existing hostels.

3.3 LITERATURE SURVEY

1. Paper Name:

Design and implementation of hostel management system using java and mysql

Author:

Pooja Nagpal, Nishant Goel, Shivam Sangwan, Himanshi Dixit

Abstract:

E- The Hostel Management System framework is a software which is design to provide the facilities to the staff members as well as the students that saves the time that required by those paper works. As many students willing to live at hostel for studies the number of hostel buildings are increased that needs to be handle smartly by using the web application that decreases the stress or strain to the authorities. This application requires students and administrator's login details to take them over the application dashboard where they all can easily access the information regarding their registration for hostel rooms, fee payments, can check student records as well as allow to update whenever required. This application overcomes the drawbacks of the past methods of management system; it is user friendly,

GUI interface or environment, reliable and secured with best IT department professionals. Working on such application brings transparency in the environment that builds the trust between students and management as well as trust on the digital world.

2.Paper Name:

Hostel Management System and Aggregation

Author:

Prof. Shyamsundar Magar, Mr. Rohit Jadhav ,Ms. Sakshi Said ,Mr. Shashikant Jadhav

Abstract:

Hostel Management is a web-application which is created for booking the hostels for individual. This will limit manual work and also make hostel allocation so much easier for students and hostel administrators. It manages data in database and retrieve it whenever required. We design this system on the request of the hostel management, through this they cannot require people to handle and calculate the things. The developed system overcomes the drawbacks of manual hostel management its said to be more user-friendly, GUI oriented, reliability, efficient and secured with access control mechanisms. The old techniques with limitations have impacted educational systems.

Hence, we are developing one. Technologies like bootstrap will be used. Considering all the requirement as per hostel need Web app will be build which will work on both IOS and Android. An Hostel Management is a web-application which is created for booking the hostels for individual. This will limit manual work and also make hostel allocation so much easier for students and hostel administrators. It manages data in database and retrieve it whenever required. We design this system on the request of the hostel management, through this they cannot require people to handle and calculate the things. The developed system overcomes the drawbacks of manual hostel management its said to be more user-friendly, GUI oriented, reliability, efficient and secured with access control mechanisms. The old techniques with limitations have impacted educational systems.

Hence, we are developing one. Technologies like bootstrap will be used. Considering all the requirement as per hostel need Web app will be build which will work on both IOS and Android. Droid.

3. Paper Name:

A Study on Shift towards Digitization of Hostel Room Allotment for a University

Author: -

Tanmay Nandanwar, Priyanka Bahutule, Raviteja Buddala

Abstract: -

The paper addresses the general problems faced by students in big universities or colleges during the room allotment process. The present work aims to develop an online room allocation process over a conventional room allotment process held in several big universities or colleges. Hostel room allotment process is for the students who are willing to stay in hostel. Rooms are allotted to the students based on their merit and room preferences. The conventional room allotment process has many disadvantages. They consume more time and do not provide proper room selection on time etc. The present research work aims to tackle the general problems and depicts the implementation using a digital platform. The present work shows the design and digitalized/online room allotment web portal through which students can eliminate these problems and increase the ease of room allotment process. This project concludes with discussions on improvements and completion of a fully working website with its implementation in a university.

4. Paper Name: -

Hostel Management System (HMS)

Author: -

Prof. Deepali Narkhede¹, Rutuja Bangude², Mayuri Sonawane³, Mandar Shevade⁴

Abstract: -

This is a design and implementation of an online Hostel Management System. “ONLINE HOSTEL MANAGEMENT SYSTEM” is software developed for managing various activities in the hostel. For the past few years, the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the

system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system

5.Paper Name: -

Implementation Of Hostel Management with Automation Using Design Thinking

Author: -

Dinesh.B1, Gogul Nithin.R2, Pavatharani.R3, Sneha.R4,C.Senthilkumar

Abstract: -

“HOSTEL MANAGEMENT SYSTEM” is software developed for managing various activities in the hostel. For the past few years the numbers of educational institutions are increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a hostel and avoid the problems which occur when carried manually. This project is expected to limit human works and make hostel allocation much easier for student and hostel administrators with the help of this software. In this way, the quantity of hostels is additionally expanding for the settlement of the student considering in this college. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which more users friendly.

6. Paper Name: - Hostel management System based on Finger Print Authentication

Author: -

G. Rajkumar and T. Sivagama Sundari

Abstract: -

The biometric system plays the most important role in this current century. Finger print identification is one among the foremost distinguished and familiar identity verification system due to its individuality. Security within the hostel is one of the foremost repetitive issues. To keep up day by day attendance verification is sophisticated and time-consuming system for the hostel management. There are number of existing attending systems are available for college students, for hostel students it must improve. Within the existing system wardens are manually maintain the attendance for hostel students. This paper deals with,

avoid of an entire problem in hostel management system together with this monitoring system also proposed. The administrator of this system was college principal or warden. Biometric system is used to accommodate a large number of students within the hostel. This system makes automatically to monitor the entry and exit of students from hostel and offers alert SMS to parents for their safety.

7. Paper Name: - Study of Digitalized Hostel Management System

Author: -

Kartik Chaudhri, Riddhi Kevat

Abstract: -

“E-Hostel” is an android application to automate the manual hostel management system of particular hostel associated with collage. It solves the important problems like room allocation, fee and fee receipt management, student management, complaint management, maintain visitor records, student leave management, and notice board. This application provides some additional facilities like secure user authentication, digital hostel pass, live notification, important form updates, user profile management.

8. Paper Name: - Online Hostel Management System for Sanskrithi School of Engineering

Author: -

P. Meghana D. Jyothi E. Jahanvi

Abstract: -

“HOSTEL MANAGEMENT SYSTEM” is software developed for managing various activities in the hostel. For the past few years the number of educational institutions are increasing rapidly. Thereby the number of hostels are increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and the software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problem which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more users friendly and more GUI oriented.

9. Paper Name: - Hostel Management System (HMS)

Author: -

Prof.Kesh, Prasad, Philip Wirtz

Abstract: -

This study analyzes hostel student satisfaction using an electronic guest feedback system based on web and Internet technologies. Data collected from a sample of 1,218 hostel student who stayed at two midprice full-service hostels in Orlando, Florida during a 1-year period was used for this study. Structural equation modeling was employed to identify the underlying relationship between guest evaluation of hostel facilities and staff service on the one hand, and perceived value, satisfaction, and intent to revisit and recommend on the other. It is also noted that significant moderating effects affected the student evaluations of the hostel facilities and staff service quality. Management implications include suggestions for enhancing student satisfaction with hostel operations leveraging web-based technology. This study should help hostel management focus on those aspects of the guest experience that matter most for driving satisfaction ratings.

10. Paper Name: - Hostel Intelligent Building System

Author: -

Iuliia Myts1

Abstract: -

Therefore, Intelligent Management System provides complete automation of the hotel room. Intelligent hotel room is equipped with the following devices: magnetic key holder, bedside unit installed to operate air quality system (conditioner), motorized curtains/louvers, audio-video systems and to change the display image icons of the doorbell staying in bed.

Energy supply management module of the room will switch off all the electric devices when the guest leaves the room. This will save electricity by turning off unnecessarily working electric means. Thus, having studied the issue of the Intelligent Management System the following conclusion for the

CHAPTER 4
PROBLEM STATEMENT AND SCOPE

4.1 PROBLEM STATEMENT

Design and develop a Hostel Management System to address the challenges faced by a large hostel facility in efficiently managing its operations. The hostel currently faces several manual and time-consuming processes, leading to inefficiencies, data inaccuracies, and reduced guest satisfaction. The objective is to create a comprehensive software solution that automates key tasks and enhances overall hostel management.

4.1.1 Goals and Objectives

Objectives:

Efficiency Improvement:

Streamline administrative processes such as student registration, room allocation, fee management, and maintenance requests to improve operational efficiency.

Enhanced Student Experience:

Provide a user-friendly platform for students to manage their hostel-related tasks, such as room selection, fee payments, maintenance requests, and communication with hostel staff.

Resource Optimization:

Optimize the allocation and utilization of hostel resources, including rooms, inventory, and staff, to ensure efficient operations and minimize wastage.

Transparency and Accountability:

Maintain transparent and accessible records of all hostel activities, transactions, and communications to promote accountability among staff and students.

Data-Driven Decision Making:

Collect and analyze data on hostel occupancy, finances, and student preferences to support informed decision making by hostel administrators.

Scalability and Flexibility:

Design a modular and scalable system that can easily adapt to changes in student enrollment, hostel facilities, and administrative requirements.

Improved Communication:

Facilitate effective communication between hostel staff, students, and administrators through automated notifications, alerts, and a centralized platform for managing interactions.

Goals:**Automation of Administrative Tasks:**

Automate routine administrative tasks, such as student registration, room allocation, fee collection, and maintenance request handling, to reduce manual workload and minimize errors.

Enhanced User Experience:

Provide a seamless and intuitive user experience for both students and hostel staff, with easy access to relevant information, quick response times, and personalized services.

Real-Time Monitoring and Reporting:

Enable real-time monitoring of hostel occupancy, resource utilization, financial transactions, and maintenance activities, with customizable reporting features to track key metrics and trends.

Secure and Accessible Platform:

Implement robust security measures to protect sensitive data and ensure secure access to the system for authorized users, while maintaining accessibility for all stakeholders.

Integration with Existing Systems:

Integrate the Hostel Management System with other existing college systems, such as the student information system and accounting software, to streamline data exchange and ensure consistency across platforms.

Continuous Improvement and Feedback Loop:

Establish mechanisms for gathering feedback from users, analyzing system performance, and

incorporating enhancements and updates based on user needs and evolving requirements.

Compliance with Regulations and Standards:

Ensure compliance with relevant regulations, policies, and industry standards governing hostel management, data privacy, and security, to maintain trust and credibility among stakeholders.

Positive Impact on Operations and Student Life:

Measure the impact of the Hostel Management System on hostel operations, student satisfaction, and overall campus life, with the goal of continuously improving the hostel experience for all residents.

4.1.2 Statement of scope

Project Title: Hostel Management System

1. Project Description:

- The Hostel Management System project aims to develop a comprehensive, user-friendly digital platform to automate and streamline hostel operations for a college hostel. The system will serve as a centralized hub for managing administrative tasks, student interactions, resource allocation, and communication, ultimately enhancing efficiency, transparency, and the overall hostel experience for both students and staff.

2. Project Deliverables:

- Software application: This is the core deliverable - a fully functional hostel management system application. It should encompass all the features and functionalities defined in your project plan. This could be a web-based application, a mobile application, or a combination of both.
- User manuals and documentation: Create comprehensive user manuals for both hostel staff and students. These manuals should explain how to navigate the system, use its various features, and troubleshoot any issues.
- Sample data: Populate the system with sample data for testing purposes. This allows users to explore the functionalities and get familiar with the system before using real data.

3. Project Inclusions:

- Face Student Management:
- Room Management:
- Fee Management:
- Maintenance and Service Requests:
- Visitor Management:
- Inventory and Asset Management:
- Communication Tools:
- Security Features:

4. Project Exclusions:

- Third-Party Integrations
- Customization Beyond Scope
- Hardware Procurement and Installation
- Extensive User Training
- Support for Outdated Technologies
- Continuous System Maintenance

5. Assumptions and Dependencies:

- Availability of Stakeholder Input:
- Access to Required Data:
- Compliance with Regulations:
- Standardized Processes:
- Internet Connectivity:
- User Training and Adoption:
- Infrastructure Readiness:

6. Success Criteria:

- The face liveliness detection system successfully integrates with the learning management system with face liveliness detection.
- Face detection and basic liveliness checks achieve acceptable accuracy levels.
- User feedback indicates a positive experience with the system's ease of use and effectiveness in preventing cheating.

7. Project Timeline:

The estimated development and integration timeline is [N] weeks, subject to change based on unforeseen challenges.

8. Project Team:

The project team will consist of individuals with expertise in front end technologies , software development, database management.

4.2 MAJOR CONSTRAINTS

Budget Constraints:

Limited budget allocation for the project, which may restrict the scope of features, technology choices, or resources available for development.

Time Constraints:

Tight project deadlines or timeframes, which may limit the time available for requirement gathering, design, development, testing, and deployment phases.

Resource Constraints:

Limited availability of skilled resources such as developers, designers, testers, and project managers, which may impact the project's progress and delivery timeline.

Scope Constraints:

Defined scope of the project may restrict the inclusion of certain features or functionalities that stakeholders desire, leading to trade-offs and prioritization of requirements.

Technical Constraints:

Technical limitations or dependencies, such as compatibility issues with existing systems, infrastructure constraints, or limitations of chosen technologies.

Regulatory Constraints:

Compliance requirements with legal regulations, data protection laws, and institutional policies, which may impose constraints on data handling, security measures, and user privacy.

User Adoption Constraints:

Resistance to change or lack of user buy-in from hostel staff and residents, which may hinder the successful adoption and usage of the system.

Infrastructure Constraints:

Limitations in infrastructure resources such as server capacity, network bandwidth, or internet connectivity, which may affect system performance and scalability.

Integration Constraints:

Challenges in integrating the hostel management system with existing college/university systems, third-party services, or external databases, leading to data synchronization issues or interoperability challenges.

Security Constraints:

Security vulnerabilities and risks, including data breaches, unauthorized access, or data loss, which may require robust security measures and protocols to mitigate.

Maintenance and Support Constraints:

Limited availability of resources and budget for post-deployment maintenance, support, and updates, which may impact the long-term sustainability and usability of the system.

4.3 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

Efficiency Issues in Our Hostel Management System:

Time Management

- Problem: Inefficient time use in administrative tasks and service delivery.
- Solution: Implement scheduling tools and automate repetitive tasks.

Resource Allocation

- Problem: Inefficient use of resources like staff, rooms, and utilities.
- Solution: Use management software to optimize resource allocation and monitor usage.

Communication Barriers

- Problem: Poor communication among staff and residents can lead to misunderstandings and delays.
- Solution: Establish clear communication channels and use collaborative tools.

Lack of Clear Policies and Procedures

- Problem: Ambiguous policies can result in inconsistent service delivery.
- Solution: Develop and document clear policies and procedures, and ensure they are communicated effectively.

Maintenance and Upkeep

- Problem: Delayed maintenance can lead to larger, more costly issues.
- Solution: Implement a preventive maintenance schedule and use a tracking system for maintenance requests.

Data Management

- Problem: Inefficient handling of data related to residents, bookings, and payments.
- Solution: Implement a robust hostel management system that centralizes and automates data -handling.

Components:

Interface (UI):

- **Dashboard:** Provides an overview of hostel activities, such as room occupancy, maintenance requests, and fee collections.
- **Student Portal:** Allows students to access their profiles, view room assignments, submit maintenance requests, and make fee payments.
- **Admin Panel:** Enables administrators to manage hostel facilities, allocate rooms, track fees, and generate reports.

Database Management:

- **Student Database:** Stores information about hostel residents, including personal details, contact information, and room assignments.
- **Room Inventory:** Maintains records of available rooms, room types, occupancy status, and maintenance history.
- **Fee Management:** Stores data related to fee structures, payment transactions, outstanding balances, and payment histories.

Room Allocation and Management:

- **Room Assignment:** Automates the allocation of rooms based on availability, and administrative rules.
- **Room Inventory Management:** Tracks room occupancy, availability, room condition, and maintenance schedules.
- **Check-in/Check-out:** Manages the check-in and check-out processes for students, updating room statuses accordingly.

Fee Management:

- **Fee Structure Setup:** Defines fee components, payment schedules, and billing cycles for hostel fees.
- **Fee Collection:** Facilitates fee collection through various payment methods, such as cash, online payments, or bank transfers.
- **Fee Tracking:** Monitors fee payments, overdue balances, payment reminders, and financial reporting.

Maintenance and Service Requests:

- **Maintenance Requests:** Allows students to submit maintenance requests for issues in their rooms or common areas.
- **Task Assignment:** Assigns maintenance tasks to staff members, tracks task statuses, and ensures timely resolution.
- **Maintenance History:** Maintains records of maintenance activities, including requests, repairs, and maintenance schedules.

Communication Tools:

- **Internal Messaging:** Facilitates communication between hostel staff, students, and administrators through messaging features.
- **Announcements and Notifications:** Sends notifications and alerts to users regarding important updates, events, or announcements.

Reporting and Analytics:

- **Reporting Dashboard:** Generates reports on key hostel metrics, including occupancy rates, fee collections, maintenance activities, and financial summaries.
- **Analytics Tools:** Provides data analysis and visualization tools to identify trends, patterns, and areas for improvement in hostel management.

Security Features:

- **Access Control:** Implements role-based access control (RBAC) to restrict access to sensitive information and functionalities based on user roles.
- **Data Security:** Ensures data security through encryption, access controls, and compliance with data protection regulations.

Integration with External Systems:

- **Integration APIs:** Provides APIs for integration with college/university databases, accounting systems, or third-party services for data exchange and synchronization.

System Flow:

1. User Authentication and Authorization

User Login:

Users (students, administrators, and staff) log into the system using their credentials.

System verifies credentials and grants access based on user roles.

2. Student Registration and Profile Management

New Student Registration:

Students fill out an online registration form with personal details, contact information, and emergency contacts.

Administrators verify and approve the registration.

System creates a student profile and assigns a unique student ID.

Profile Management:

Students can view and update their profiles as needed.

Changes are saved and updated in the system database.

3. Room Allocation and Management

Room Allocation:

Students submit room preferences during registration or room change requests.

Administrators allocate rooms based on preferences, availability, and predefined rules (e.g., gender, course year).

System updates room occupancy status and assigns the room to the student.

Room Management:

Administrators can reassign rooms, manage room conditions, and schedule maintenance.

System tracks room status (occupied, vacant, under maintenance).

4. Fee Management

Fee Structure Setup:

Administrators define fee structures, including room rent, utilities, and additional charges.

System generates invoices based on the defined fee structure.

Fee Collection:

Students can view their fee details and make payments through various methods (online, bank transfer, cash).

System records payments and updates the payment status.

Fee Tracking and Reminders:

System tracks due payments and sends automated reminders to students for pending fees.

Administrators can generate reports on fee collections and outstanding balances.

Methodologies:

- Agile Methodology:

Agile is an iterative and incremental approach that emphasizes flexibility, collaboration, and customer feedback. It's particularly useful when requirements are expected to evolve.

- Key Practices:

Scrum Framework: Utilize sprints (time-boxed iterations) of 2-4 weeks, with daily standups to monitor progress.

User Stories: Requirements are captured as user stories, focusing on delivering small, functional pieces of the system.

Backlog Management: Maintain a prioritized backlog of tasks and features.

Sprint Planning and Reviews: Plan each sprint, review progress at the end, and gather feedback for continuous improvement.

Retrospectives: Reflect on the sprint to identify what went well and what could be improved.

Benefits:

Flexibility to accommodate changing requirements.

Frequent delivery of functional software.

Continuous stakeholder engagement and feedback.

Efficiency Issues and Solutions:

- Database Performance:
 - Slow database queries can cause delays in retrieving and updating information.
 - Data redundancy and poor indexing can lead to inefficient data retrieval.
- Scalability:
 - The system may struggle to handle increasing numbers of users and data entries.
 - Inefficient handling of concurrent users can lead to performance bottlenecks.
- Resource Management:
 - Inefficient algorithms for room allocation and resource scheduling can result in suboptimal use of resources.
- System Load and Response Time:
 - High load times and slow response times can degrade the user experience.
 - Poorly optimized code and lack of caching mechanisms can contribute to this issue.
- Integration with External Systems:
 - Lack of seamless integration with other systems (e.g., university databases, payment gateways) can lead to data inconsistencies and manual workarounds.
- Security:
 - Vulnerabilities in the system can lead to unauthorized access and data breaches.
 - Inadequate encryption and access control measures can compromise data security.

Solutions:

- Optimizing Database Performance:
 - **Indexing:** Create appropriate indexes on frequently queried fields to speed up data retrieval.
 - **Query Optimization:** Rewrite complex queries for better performance and use query profiling tools to identify slow queries.

- **Normalization:** Normalize the database schema to eliminate redundancy and ensure efficient storage.
- **Database Sharding:** Distribute data across multiple databases (shards) to handle large volumes of data and improve query performance.

Enhancing Scalability:

- **Load Balancing:** Implement load balancers to distribute incoming traffic across multiple servers, ensuring no single server is overwhelmed.
- **Horizontal Scaling:** Add more servers to the system to handle increased load, rather than relying on a single powerful server.
- **Microservices Architecture:** Break down the system into smaller, independent services that can be scaled individually based on demand.

Improving Resource Management:

- **Efficient Algorithms:** Develop and implement efficient algorithms for room allocation, maintenance scheduling, and staff management. Use optimization techniques like linear programming.
- **Real-time Monitoring:** Implement real-time monitoring and dynamic adjustment of resources based on current usage patterns and needs.

Reducing System Load and Improving Response Time:

- **Caching:** Use caching strategies (e.g., in-memory caching with Redis or Memcached) to store frequently accessed data and reduce database load.
- **Asynchronous Processing:** Implement asynchronous processing for non-critical tasks to improve response times for user-facing operations.
- **Code Optimization:** Profile and optimize application code to remove bottlenecks and improve execution efficiency.

Seamless Integration with External Systems:

- **APIs:** Develop robust APIs for integration with external systems such as university

databases, payment gateways, and identity management systems.

- **Middleware:** Use middleware to facilitate communication between different systems and ensure data consistency.
- **Data Synchronization:** Implement data synchronization mechanisms to keep data consistent across integrated systems in real-time.

Enhancing Security:

- **Encryption:** Use strong encryption (e.g., AES-256) for sensitive data both in transit (TLS/SSL) and at rest.
- **Access Control:** Implement fine-grained access control using role-based access control (RBAC) or attribute-based access control (ABAC).
- **Regular Security Audits:** Conduct regular security audits and vulnerability assessments to identify and address potential security risks.

User Interface Optimization:

- **Focus on User Needs:**

Identify your users: Consider the different user groups (students, hostel staff, administrators) and their specific needs when designing the UI.

User research: Conduct user research through surveys, interviews, or usability testing to understand user pain points and preferences.

- **Design Principles for Optimization:**

Simplicity and clarity: Prioritize a clean, uncluttered interface with clear labels, icons, and navigation menus. Users should be able to intuitively understand where to find what they need.

Accessibility: Ensure the UI is accessible for users with disabilities. Use good color contrast, offer alternative text descriptions for images, and consider keyboard navigation options.

4.4 OUTCOME

The outcome of a Hostel Management System project should provide significant improvements in the management and operational efficiency of a hostel. Here is a detailed overview of the expected outcomes:

Key Outcomes

- Streamlined Operations
- Enhanced User Experience
- Improved Resource Management
- Efficient Communication
- Data Integrity and Security
- Scalability and Flexibility
- Comprehensive Reporting and Analytics
- Cost Savings

Detailed Outcomes

1. Streamlined Operations

- **Automated Processes:** Tasks such as room allocation, fee collection, and maintenance requests are automated, reducing the workload on administrative staff.
- **Centralized Management:** A single system to manage all aspects of hostel operations, from admissions to daily operations, leading to better coordination and efficiency.

2. Enhanced User Experience

- **User-Friendly Interface:** An intuitive and responsive UI that simplifies navigation and use for both staff and students.
- **Self-Service Portal:** Students can easily perform tasks such as checking room availability, making payments, and submitting maintenance requests online.
- **Real-Time Access:** Users can access the system from anywhere, providing real-time updates and services.

3. Improved Resource Management

- **Optimized Room Allocation:** Efficient algorithms ensure optimal allocation of rooms based on availability and student preferences.
- **Maintenance Scheduling:** Automated scheduling and tracking of maintenance tasks to ensure timely completion and reduce downtime.
- **Inventory Management:** Better management of hostel resources and supplies, ensuring availability when needed.

4. Efficient Communication

- **Integrated Messaging System:** Facilitates communication between students, staff, and administration through in-system messaging and notifications.
- **Notifications and Alerts:** Automated alerts and notifications for important events such as fee due dates, maintenance schedules, and announcements.

5. Data Integrity and Security

- **Secure Data Storage:** Implementation of robust security measures to protect sensitive student and administrative data.
- **Access Control:** Role-based access control ensures that only authorized personnel have access to specific data and functions.
- **Data Backup:** Regular data backups to prevent data loss and ensure continuity.

6. Scalability and Flexibility

- **Scalable Architecture:** The system is designed to scale easily, accommodating increasing numbers of students and data without performance degradation.
- **Modular Design:** Flexibility to add new features and modules as needed without major overhauls.

7. Comprehensive Reporting and Analytics

- **Detailed Reports:** Generate comprehensive reports on various aspects of hostel management, including occupancy rates, fee collection, and maintenance activities.
- **Data Analytics:** Use analytics to identify trends, predict future needs, and make informed decisions.

- **Dashboard:** An executive dashboard providing at-a-glance views of key metrics and KPIs.

8. Cost Savings

- **Reduced Manual Work:** Automation reduces the need for extensive manual labor, cutting down on administrative costs.
- **Optimized Resource Use:** Efficient resource management leads to cost savings on utilities, supplies, and maintenance.
- **Improved Planning:** Better data and analytics enable more accurate budget planning and resource allocation.

4.5 APPLICATIONS

- **Students:**

Improved living experience: Self-service features like online booking, fee payment, and maintenance requests offer convenience and reduced stress.

Increased transparency: Access to real-time information on room availability, fees, and hostel rules promotes informed decision-making.

Enhanced communication: A dedicated communication channel within the HMS facilitates interaction with hostel staff and fosters a sense of community.

Accessibility features: Features like multilingual support, screen reader compatibility, and text-to-speech options can cater to students with diverse needs.

- **Hostel Staff:**

Streamlined operations: Automation of tasks like room allocation, booking management, and fee collection significantly reduces workload.

Improved data management: Centralized data storage ensures accuracy, facilitates reporting, and allows for better decision-making.

Enhanced communication: Efficient communication channels within the system improve responsiveness to student needs and inquiries.

Data-driven insights: Reports generated by the HMS can identify areas for improvement in resource allocation, maintenance scheduling, and overall hostel operations.

- **College Administration:**

Cost savings: Automation and improved efficiency can lead to reduced operational costs for the college.

Improved reputation: A well-run hostel with a user-friendly HMS can attract students and contribute positively to the college's image.

Data for future planning: Data collected by the HMS can inform future decisions about hostel expansions, renovation projects, or student needs assessment.

- **Beyond the College:**

Parents/Guardians: Increased transparency and ease of communication regarding their child's hostel experience can provide peace of mind.

Local Businesses: Information about events or services offered within the hostel can be displayed on the HMS, potentially attracting local businesses seeking student customers.

Government Agencies: Data on student demographics and hostel occupancy rates can be helpful for government agencies planning student housing initiatives.

4.6 HARDWARE SOURCE REQUIREMENT

Sr. No	Parameter	Minimum Requirements	Justification
1	Processor	Dual-core CPU (2.0 GHz or higher)	Efficient face detection and basic liveliness checks using Haar cascades.
2	RAM	4 GB	Smooth system operation and multitasking during assessments.
3	Webcam	480p resolution	Capture clear facial images of students for accurate detection.
4	Storage	100 GB free space	Stores data and potentially captured images (adhere to privacy regulations).
5	Internet Connection	Stable broadband connection (10 Mbps or higher)	Real-time communication with the LMS for authentication and data exchange.

Table 4.1: Hardware Requirement

4.7 SOFTWARE RESOURCES REQUIRED

Sr. No	Software	Description	Justification
1	Operating System	Windows 11, macOS 12 (or later versions)	Widely used platforms with good hardware compatibility.
2	Anaconda	Python distribution with scientific libraries	Provides a pre-configured environment with essential libraries for computer vision (OpenCV) and potentially machine learning (scikit-learn) if needed.
3	Spyder	Integrated Development Environment (IDE) for Python	User-friendly interface for coding, debugging, and data exploration within Anaconda.
4	OpenCV package within Anaconda	Open-source library with pre-trained Haar cascades for face detection.	Included in Anaconda for convenient installation and use.
5.	Additional packages within Anaconda	dlib (facial landmark detection), scikit-learn (machine learning), pillow, TensorFlow, keras	Can be installed within Anaconda for advanced liveliness detection techniques.

Table 4.2: Software Requirements

CHAPTER 5
PROJECT PLAN

5.1 PROJECT ESTIMATES

Effort Estimation:

Effort estimation for a Hostel Management System (HMS) involves estimating the total amount of work required from the project team to complete the development, testing, and deployment of the system. Here's a general approach to estimating effort for an HMS project:

1. Define Work Breakdown Structure (WBS):

- Requirement Gathering: Gather and document functional and non-functional requirements for the HMS.
- Design: Create design specifications for the HMS architecture, database schema, user interface, etc.
- Development: Code the HMS modules based on the design specifications.
- Testing: Test the HMS for functionality, usability, performance, security, and compatibility.
- Deployment: Deploy the HMS to production servers and ensure it is accessible to users.
- Maintenance: Provide ongoing support, updates, and bug fixes for the HMS post-deployment.

2. Estimate Effort for Each Task:

- Requirement Gathering: Estimate the effort required for stakeholder meetings, documentation, and requirements analysis.
- Design: Estimate the effort for designing the system architecture, database schema, UI wireframes, etc.
- Development: Estimate the effort for coding, unit testing, integration, and debugging of the HMS modules.
- Testing: Estimate the effort for test planning, test case development, test execution, and defect management.
- Deployment: Estimate the effort for server setup, configuration, deployment, and user training.
- Maintenance: Estimate the ongoing effort for handling support requests, applying updates, and addressing bugs.

Factors Affecting Effort:

Several factors can influence the effort required for developing a Hostel Management System (HMS). Here are some key factors to consider:

1. Scope and Complexity of Requirements:

The scope and complexity of functionalities required in the HMS play a significant role in determining the effort. A more extensive set of features or complex requirements may require additional time and resources to implement.

2. Technology Stack:

The choice of technology stack, including programming languages, frameworks, and databases, can impact the effort required. A familiar technology stack for the development team may result in faster development, while adopting new technologies may require additional learning and implementation time.

Resource Estimation:

- **Personnel:** Estimate the number of developers, testers are required based on the project scope and PM effort.
- **Hardware:** The hardware requirements (refer to previous section) are relatively modest for a basic implementation. Consider potential upgrades if needed for a larger user base.
- **Software:** Anaconda with Spyder offers a cost-effective development environment. Additional library licenses might be required for advanced features

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

Effort Estimation and Cost Breakdown (Minimal Cost)

Effort per Team Member:

Considering a basic demo and aiming for minimal cost, we'll target a 1 Person-Month (PM) effort per team member, which translates to approximately 40 hours of work per person.

Total Effort (Weeks):

Since we're provided with a 10-month timeframe, and a month has roughly 4.3 weeks on average, we can estimate the total project duration in weeks as follows:

Total Project Duration (Weeks) = Months x Weeks/Month

Total Project Duration (Weeks) = 10 months x 3.1 weeks/month (approximately)

Total Project Duration (Weeks) = 31 weeks

Effort Cost Estimation:

To minimize costs, we'll use a conservative estimate of ₹120 per hour for student developer rates in India.

Effort Cost per Person (Weeks):

We can express the effort cost per person in weeks by dividing the total working hours by the hourly rate:

Effort Cost per Person (Weeks) = Working Hours (hours) / Hourly Rate (₹/hour)

Effort Cost per Person (Weeks) = 40 hours / ₹150/hour

Effort Cost per Person (Weeks) = Approximately 0.27 weeks (This represents the number of weeks a single person would work at ₹150/hour to complete 40 hours of work)

Overall Project Cost:

Our goal is to keep the overall project cost around ₹15,000. With a team size of 4 and the effort cost per person being approximately 0.27 weeks, we can estimate the total team effort cost in weeks

Total Team Effort Cost (Weeks) = Team Size (People) x Effort Cost per Person (Weeks)
Total Team Effort Cost (Weeks) = 4 People x 0.27 weeks/Person
Total Team Effort Cost (Weeks) = Approximately 1.08 weeks (This represents the combined effort of all team members based on the individual effort cost per person)

Translating Effort Cost into Monetary Value:

To arrive at a total cost in rupees, we can multiply the total team effort cost in weeks by the chosen hourly rate:

Total Project Cost (₹) = Total Team Effort Cost (Weeks) x Hourly Rate (₹/hour) x Working Hours per Week
Total Project Cost (₹) = 1.08 weeks x ₹150/hour x 40 hours/week (assuming a standard 40-hour work week)
Total Project Cost (₹) = Approximately ₹16,200

Breakdown:

- Team Size: 4 People
- Effort per Person: 1 Person-Month (40 hours)
- Effort Cost per Person (Weeks): 0.27 weeks (approximately)
- **Overall Project Cost:** ₹13,200 (approximately)

5.1.1.2 Time Estimates

Project: Hostel Management System

Factors Affecting Time:

- Technology Stack:

The choice of technology stack can affect development time. Familiarity with the chosen technologies and frameworks may expedite development, while learning and implementing new technologies may extend the project timeline.

- Team Composition and Expertise:

The size, skill levels, and experience of the project team play a crucial role in project duration. A well-qualified and experienced team may complete tasks more efficiently, whereas a less experienced team may require additional time for learning and training.

- Project Size and Scale:

The size and scale of the HMS project, including the number of users, hostel facilities, and administrative processes to be managed, can influence the project timeline. Larger projects typically require more time due to their complexity and scale.

- Requirements Changes:

Frequent changes or additions to the requirements during the development process can extend the project timeline. It's essential to manage requirements effectively to minimize scope creep and avoid unnecessary delays.

- Integration Requirements:

Integration with existing systems or third-party services may add complexity to the project and increase development time. Compatibility issues, API integrations, and data migrations should be carefully planned and executed.

- Testing and Quality Assurance:

Thorough testing is essential to ensure the functionality, usability, and reliability of the HMS. Adequate time should be allocated for test planning, test case development, test execution, and defect management, which can impact the overall project timeline.

- **Project Management and Communication:**

Effective project management practices, clear communication channels, and collaboration among team members can streamline the development process and reduce delays. Poor project management or communication can lead to misunderstandings, bottlenecks, and rework, extending the project timeline.

- **Resource Availability:**

The availability of resources such as developers, testers, and infrastructure can affect project timelines. Adequate resource allocation and efficient resource management are essential to optimize project duration.

- **External Dependencies:**

Dependencies on external factors such as vendors, regulatory approvals, or client feedback can impact project timelines. It's crucial to identify and manage these dependencies proactively to avoid delays.

5.1.2 Project Resources

- **Hardware Resources:**

Server Infrastructure:

- Hardware components such as servers, storage devices, and networking equipment required for hosting the HMS application and database.
- Client Devices: Computers, laptops, tablets, and smartphones used by users (administrators, staff, and students) to access the HMS application.

Software Resources:

- Development Tools: Integrated Development Environments (IDEs), version control systems, project management tools, and collaboration platforms used by the development team.
- Database Management System (DBMS): Software for managing and interacting with the HMS database, such as MySQL, PostgreSQL, or MongoDB.

- **Documentation and Communication Resources:**

Project Documentation:

- Templates, guidelines, and tools for creating project plans, requirement documents, design specifications, and user manuals.
- Communication Tools: Email, instant messaging, video conferencing, and project management platforms used for communication and collaboration among team members and stakeholders.

Training and Knowledge Resources:

- Training Materials: Documentation, tutorials, and training sessions to familiarize team members with the HMS requirements, technologies, and tools.
- External Training and Courses: Access to online courses, workshops, and seminars to enhance the skills and knowledge of team members as needed.

External Services and Support:

- Cloud Services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS) offerings used for hosting, deployment, and other cloud-based services.
- Technical Support: Access to technical support services provided by vendors or third-party providers for resolving issues or addressing technical challenges during the development and deployment of the HMS.

5.2 RISK MANAGEMENT W.R.T.NP HARD ANALYSIS

1. Identifying Risks:

- Complexity of Algorithms:
 - Identifying algorithms or optimization problems within the HMS that are NP-hard or computationally complex.

- Recognizing potential challenges in solving NP-hard problems efficiently, especially when dealing with large datasets or resource constraints.

- **Algorithmic Optimization:**

- Identifying areas within the HMS where algorithmic optimization is crucial for performance and scalability.
- Recognizing the risk of suboptimal performance or scalability issues if NP-hard problems are not addressed effectively.

2. Assessing Risks:

- **Impact of NP-Hard Problems:**

- Assessing the potential impact of NP-hard problems on system performance, responsiveness, and scalability.
- Evaluating the risk of increased computational complexity leading to slower response times or system bottlenecks.

- **Resource Requirements:**

- Assessing the resource requirements (e.g., CPU, memory, storage) for solving NP-hard problems within the HMS.
- Evaluating the risk of resource constraints affecting the feasibility or efficiency of algorithmic solutions.

3. Mitigating Risks:

- **Algorithm Selection:**

- Mitigating the risk of NP-hard complexity by carefully selecting algorithms or approximation techniques suited to the problem domain.
- Exploring heuristic or approximation algorithms that provide near-optimal solutions within acceptable time frames.

- **Performance Optimization:**
 - Mitigating performance risks by optimizing algorithms, data structures, and implementation techniques.
 - Implementing caching mechanisms, parallel processing, or distributed computing strategies to improve performance for NP-hard problems.

- **Scalability Planning:**
 - Mitigating scalability risks by designing the HMS architecture to handle increasing computational demands.
 - Implementing scalable infrastructure and adopting technologies that support horizontal scaling to accommodate growing workloads.

- **Testing and Validation:**
 - Mitigating risks through thorough testing and validation of algorithmic solutions for NP-hard problems.
 - Conducting performance testing, stress testing, and scenario-based testing to identify and address potential bottlenecks or limitations.

4. Contingency Planning:

- **Fallback Strategies:**
 - Developing fallback strategies or alternative approaches in case NP-hard problems cannot be solved within acceptable time frames.
 - Implementing graceful degradation mechanisms to ensure the HMS remains functional under heavy computational loads.

- **Monitoring and Feedback:**
 - Implementing monitoring tools and feedback mechanisms to track system performance and identify issues related to NP-hard complexity.
 - Establishing protocols for proactive intervention and optimization based on real-time performance data.

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set

Creating a project schedule for a Hostel Management System (HMS) involves breaking down the project into manageable tasks and allocating time for each task to ensure timely completion.

Phase 1: Planning and Preparation

Requirement Gathering:

Duration: 2 weeks

Schedule meetings with stakeholders, gather requirements, and document functional and non-functional requirements.

Team Formation and Planning:

Duration: 1 week

Form project team, assign roles and responsibilities, and create a detailed project plan.

Phase 2: Design

System Architecture Design:

Duration: 2 weeks

Design the overall architecture of the HMS, including system components, modules, and their interactions.

Database Design:

Duration: 2 weeks

Design the database schema, including tables, relationships, and data models.

User Interface (UI/UX) Design:

Duration: 2 weeks

Design the user interface for the HMS, including wireframes, mockups, and user workflows.

Phase 3: Development

Front-end Development:

Duration: 4 weeks

Implement the user interface design using HTML, CSS, JavaScript, and front-end frameworks.

Back-end Development:

Duration: 6 weeks

Develop the server-side logic, APIs, and business logic using programming languages (e.g., Python, Java, Node.js).

Database Implementation:

Duration: 3 weeks

Set up the database infrastructure, create tables, and implement data access and manipulation logic.

Phase 4: Testing

Test Planning:

Duration: 1 week

Develop a test plan outlining testing strategies, test scenarios, and testing environments.

Test Case Development:

Duration: 2 weeks

Create test cases covering functional, performance, security, and usability aspects of the HMS.

Test Execution:

Duration: 4 weeks

Execute test cases, report defects, and perform regression testing as necessary.

Phase 5: Deployment and Training

Deployment Preparation:

Duration: 1 week

Prepare for deployment, including server setup, configuration, and environment testing.

Deployment and User Acceptance Testing (UAT):

Duration: 2 weeks

Deploy the HMS to production servers, conduct UAT with end-users, and address feedback.

User Training:

Duration: 1 week

Provide training sessions for hostel staff and administrators on how to use the HMS effectively.

Phase 6: Maintenance and Support

Maintenance and Support:

Ongoing

Allocate time for ongoing maintenance, bug fixes, updates, and user support as needed post-deployment.

Additional Considerations

Buffer/Contingency:

Include buffer time for unforeseen delays, changes in requirements, or technical challenges.

Regular Review and Communication:

Schedule regular review meetings and maintain open communication channels to monitor progress and address any issues promptly.

5.3.2 Task Network

This flow diagram outlines the main functionalities and user interactions within the HMS. Each module allows users (administrators, staff, and residents) to perform specific tasks related to hostel management, such as room allocation, student management, attendance tracking, and communication.

After the core components are built, the workflow moves on to testing, deployment, and documentation phases. Finally, there's a maintenance and enhancements phase for ongoing support and improvements based on user feedback and evolving requirements.

The workflow provides a high-level overview of the project tasks and their order of execution. Each task can be further broken down into subtasks and assigned to specific team members or sprints, depending on our project management approach.

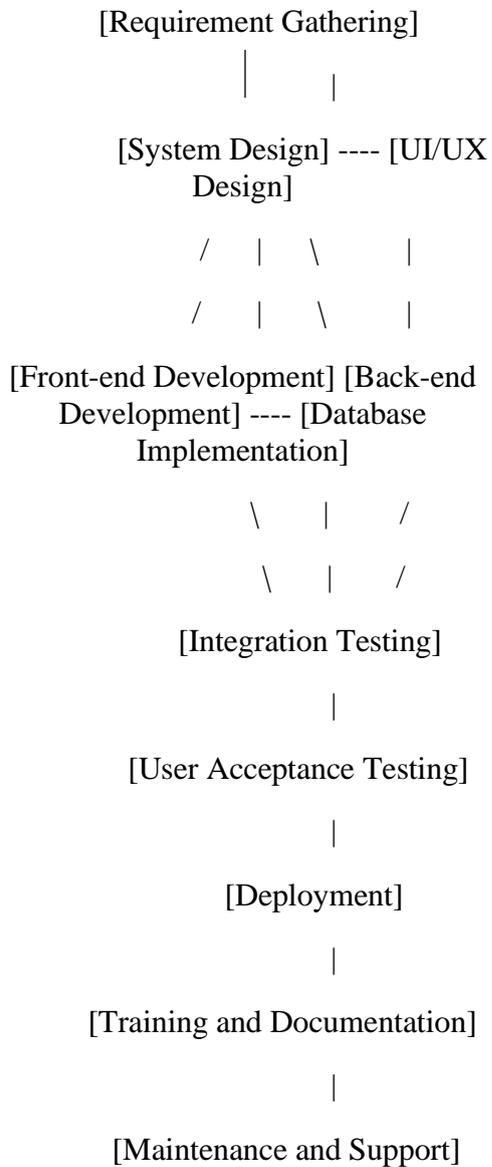


Figure 5.1: Task Network

5.4 Team Organization

We have included the responsibility for creating the overall project documentation (PowerPoint or report) and distributed it across the team:

Team Leader

- Overall project coordination and management
- Task assignment and progress tracking
- Risk management and stakeholder communication
- Ensuring timely delivery of the project
- Responsible for creating and maintaining the overall project documentation

Project Manager (Team Leader)			
Front-End Developer	Back-End Developer	AI/ML Developer	Testing and Deployment
- Front-End Documentation	- Front-End Documentation	- AI/ML Documentation	- Testing Documentation

Project Documentation
(Project Manager)

Overall Project Documentation
(Presentation / Report)
(All team members)

Front-end Developer

- Responsible for designing and developing the user interfaces
- Creating the web or desktop application interfaces for students, teachers, and administrators
- Collaborating with the back-end developer for seamless integration
- Documenting the front-end components, including UI/UX design, code structure, and usage
- Contributing to the overall project documentation (PowerPoint/report)

Back-end Developer

- Responsible for developing the server-side components
- Implementing the LMS integration and user authentication modules
- Developing APIs and handling data storage and retrieval
- Collaborating with the front-end developer and AI/ML developer for integration
- Documenting the back-end components, including APIs, database structure, and integration
- Contributing to the overall project documentation (PowerPoint/report)

AI/ML Developer

- Responsible for developing the face detection and liveness detection modules
- Researching and implementing appropriate algorithms and techniques
- Training and optimizing the machine learning models for liveness detection
- Collaborating with the back-end developer for integration
- Documenting the AI/ML components, including algorithms, models, and usage
- Contributing to the overall project documentation (PowerPoint/report)

Testing & Deployment

- Responsible for testing the entire application and its components
- Creating and executing test cases for functional, integration, and user acceptance testing
- Setting up the staging and production environments
- Deploying the application and handling deployment-related tasks
- Collaborating with other team members for testing and deployment activities
- Documenting the testing procedures, test cases, and deployment processes
- Contributing to the overall project documentation (PowerPoint/report)

5.4.1 Team Structure

Roles:

- **Project Lead (1 Student):**

- Oversees the overall project execution, ensuring adherence to deadlines and project goals.
- Manages communication within the team and facilitates collaboration.
- Tracks project progress and identifies potential roadblocks.
- Delegates tasks and ensures team members are aware of their responsibilities.
- Ideally, the project lead possesses strong communication and organizational skills.

- **Backend Developer (1 Student):**

- Focuses on developing the core functionalities related to camera integration and I checks using database for entries of student and relevant libraries.
- Possesses a strong understanding of computer vision concepts, Python programming, Database programming and Java programming

- **Frontend Developer (1 Student):**

- Responsible for integrating the UI and Interface of the system
- Utilizes available APIs or develops custom integration logic based on data documentation.
- Ensures seamless communication between the frontend and backend.
- Has experience with web development concepts (HTML, CSS, JavaScript) if user interface development is involved.

- **Tester/Documenter (1 Student):**

- Conducts unit and integration testing to identify and fix bugs within the code.
- Prepares test cases to ensure the demo functions as intended.
- Creates user testing scenarios and gathers feedback.
- Documents the codebase with comments for future reference and maintenance.
- Strong analytical and documentation skills are essential for this role.

Collaboration and Communication:

- The team should hold regular meetings (e.g., weekly) to discuss progress, address challenges, and ensure everyone is aligned.

- Utilize project management tools (e.g., Trello, Asana) to track tasks, deadlines, and communication threads.
- Encourage open communication and collaboration among team members to share knowledge and solve problems effectively.

Flexibility and Adaptation:

- This is a suggested structure, and we can adapt it based on our team's strengths and skill sets.
- If a team member possesses expertise in both computer vision and HMS development, they could handle both aspects.
- The project lead can adjust responsibilities as needed to ensure a smooth workflow.

5.4.2 Management Reporting and Communication

Regular Progress Reports:

- Provide regular updates on project status, including accomplishments, milestones achieved, and upcoming tasks.
- Frequency: Weekly or bi-weekly, depending on project timeline and stakeholder preferences.

Milestone Reports:

- Report on the completion of significant project milestones, such as system design, development phases, testing phases, and deployment.
- Include details on deliverables, timelines met, and any deviations from the original plan.

Risk and Issue Reports:

- Identify and report project risks, issues, and mitigation strategies.
- Highlight any challenges encountered during project execution and proposed solutions.

Financial Reports:

- Provide updates on project budget utilization, expenses incurred, and forecasts for remaining budget allocation.
- Include cost breakdowns for resources, tools, and other project expenses.

Quality Assurance Reports:

- Report on the results of quality assurance activities, including test coverage, defect trends, and test execution metrics.
- Provide insights into product quality and readiness for deployment.

Communication:

- Stakeholder Meetings:
- Schedule regular meetings with project stakeholders to discuss project progress, updates, and any issues or concerns.
- Ensure participation from key stakeholders, including hostel administrators, staff, residents, and project team members.

Email Updates:

- Send out email updates summarizing project progress, upcoming tasks, and important announcements.
- Include links to detailed reports and documents for stakeholders' reference.

Project Dashboards:

- Set up project dashboards or online platforms to provide real-time visibility into project status, tasks, and progress.
- Stakeholders can access dashboards to track project milestones, monitor progress, and view relevant metrics.

Document Repositories:

- Maintain centralized document repositories for project documentation, including requirements, design documents, meeting minutes, and reports.
- Ensure stakeholders have access to relevant documents and updates as needed.

Issue Tracking System:

- Use an issue tracking system to log and track project issues, enhancements, and change requests.
- Keep stakeholders informed about the status of reported issues and resolutions.

CHAPTER 6
SOFTWARE REQUIREMENT SPECIFICATION

6.1 INTRODUCTION

6.1.1 Purpose and Scope of Document

Feasibility Study/Project Proposal:

- Purpose: This document aims to convince stakeholders (e.g., college administration, investors) of the project's value.
- Scope: It outlines the problem that the HMS aims to solve (inefficiency, lack of transparency in hostel management), proposes the solution (features and functionalities of the HMS), and analyzes the project's feasibility (costs, benefits, risks).

Functional Requirements Document (FRD):

- Purpose: This document details the functionalities of the HMS for the development team.
- Scope: It outlines user stories, system features, user roles and permissions, and specific requirements for each functionality (e.g., how room booking should work, what information should be displayed on the dashboard).

Technical Design Document (TDD):

- Purpose: This document provides a technical roadmap for developers, outlining the system architecture, database design, and chosen technologies.
- Scope: It details the chosen programming languages, frameworks, and libraries, explains data structures and storage mechanisms, and defines system security protocols.

User Manual/Training Materials:

- Purpose: This document guides users (students, hostel staff) on how to interact with the HMS.
- Scope: It provides step-by-step instructions on using specific functionalities (e.g., booking a room, submitting a maintenance request), includes screenshots for clarity, and offers troubleshooting tips for common issues.

6.1.2 Overview of responsibilities of Developer

The responsibilities of a developer working on a hostel management system (HMS) can vary depending on the project's size, complexity, and chosen development approach. Here's a general overview of the key areas a developer might be involved in:

Core Responsibilities:

- Translate project requirements into a technical design document (TDD) outlining the system architecture, database structure, and chosen technologies.
- Develop the core functionalities of the HMS based on user stories and functional requirements documents (FRD). This includes writing code for features like room management, booking, fee payment, communication tools, etc.
- Integrate with external services like payment gateways, email providers, or SMS APIs (if applicable).

Testing and Debugging:

- Write unit tests to ensure individual code modules function as intended.
- Participate in system integration testing to verify different functionalities work seamlessly together.
- Debug and fix any errors or bugs identified during testing phases.

Deployment and Maintenance:

- Work with the team to deploy the HMS on a chosen hosting platform.
- Address any post-deployment issues or bugs reported by users.
- Potentially contribute to future updates and feature enhancements for the HMS.

Additional Responsibilities (Depending on Project Scope):

- **Front-End Development:** For smaller projects, the developer might handle both front-end and back-end development. This involves building the user interface (UI) of the system

using HTML, CSS, and JavaScript frameworks like React or Angular.

- Database Management: Depending on the project structure, the developer might be involved in creating and maintaining the database used by the HMS.
- API Development: If the HMS interacts with other systems or mobile applications, the developer might be responsible for developing APIs (Application Programming Interfaces) to facilitate data exchange.
- Documentation: Developers might contribute to creating technical documentation for other developers or internal teams.

Essential Skills for an HMS Developer:

- Programming languages (e.g., Python, Java, PHP)
- Database management skills (e.g., SQL)
- Understanding of web development concepts (front-end, back-end)
- Software development methodologies (Agile, Waterfall)
- Testing and debugging skills
- Problem-solving and analytical skills
- Ability to work effectively in a team

6.2 USAGE SCENARIO

- Room Selection and Booking: Sarah logs into the HMS using her student ID and password. The system displays a user-friendly interface with clear information about available room types, facilities, and pricing. She can browse photos and virtual tours of different rooms to make an informed decision.
- Online Booking and Payment: Once Sarah finds her preferred room, she can book it directly through the system. The HMS integrates with a secure payment gateway, allowing her to pay the booking fee or deposit online using her debit or credit card. This eliminates the need to wait in queues at the hostel office.
- Communication and Requests: Sarah can submit a maintenance request through the HMS if there's a problem with her room (e.g., faulty light, leaking faucet). The system allows her to attach photos or videos to clearly explain the issue. She can also track the status of her request in real-time.

- Access to Information: The HMS serves as a central hub for all hostel-related information. Sarah can access the hostel rules and regulations, view important announcements, and download forms electronically. She can also find contact details for hostel staff through the system directory.

6.2.1 User profiles

Administrator:

- Role: Overall, in charge of the hostel management system.
- Responsibilities:
 - Manage user accounts and permissions.
 - Oversee hostel operations.
 - Generate and review reports.
 - Implement policies and updates.

Permissions:

- Full access to all modules and data.
- Ability to create, read, update, and delete any records.

Responsibilities:

- Allocate rooms to students.
- Handle complaints and maintenance requests.
- Monitor hostel attendance and discipline.

Permissions:

- Access to student records and room allocation module.
- Ability to approve maintenance requests and update room status.

6.2.2 PROPOSED SYSTEM ARCHITECTURE

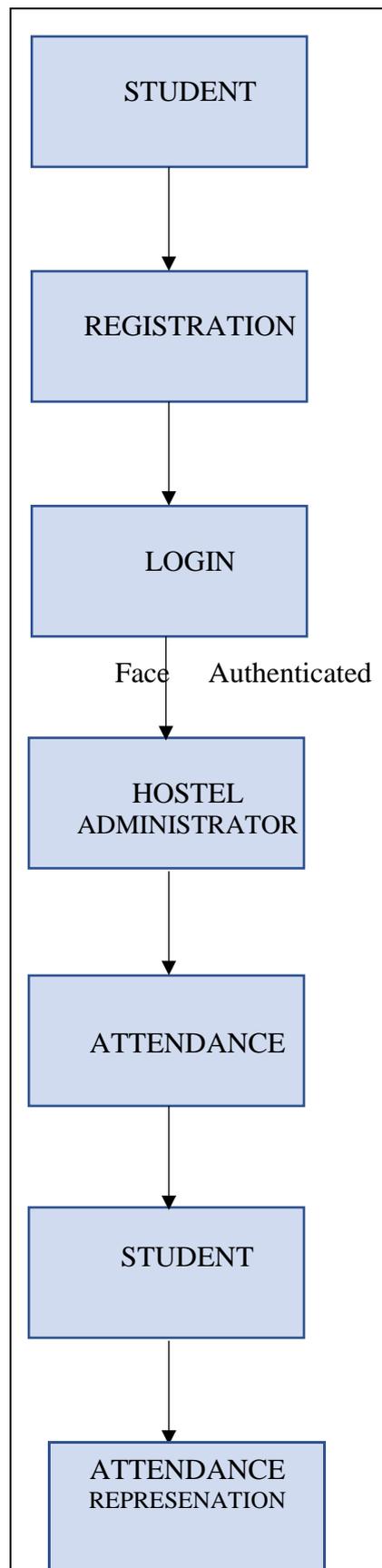


Figure 6.1: System Architecture

The Hostel Management System is a comprehensive software solution designed to streamline and manage the various aspects of hostel administration. The system aims to facilitate efficient management of room allocations, student records, maintenance requests, and other administrative tasks while ensuring a user-friendly interface for administrators, staff, and students.

6.3 Module

6.3.1 Data Flow Diagram

In Data Flow Diagram, we Show that flow of data in our system in DFD we show that base DFD in which rectangle present input as well as output and circle show our system, In DFD1 we show actual input and actual output of system input of our system is text or image and output is rumor detected like wise in DFD 2 we present operation of user as well as admin.

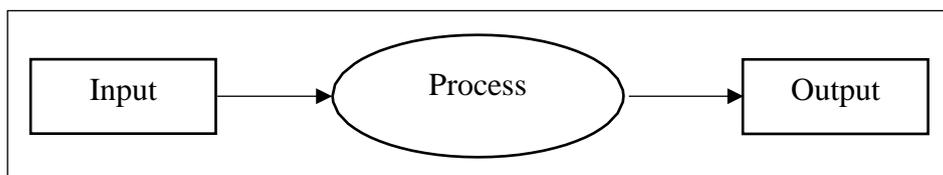


Fig 6.2: Data Flow Diagram

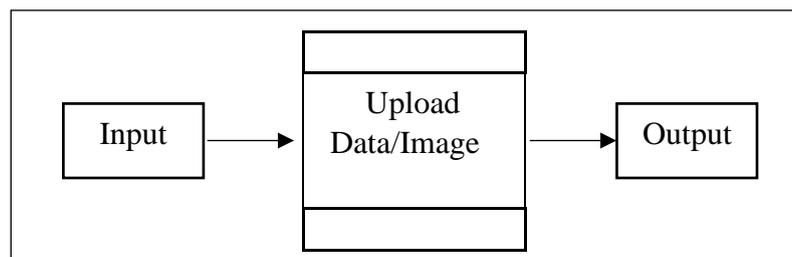


Fig 6.2: Data Flow Diagram

6.3.2 UML DIAGRAMS

- Unified Modelling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system. UML is process independent, although optimally it should

be used in process that is use case driven, architecture-centric, iterative ,and incremental. The Number of UML Diagram is available.

- Use case Diagram.
 - Component Diagram.
 - Activity Diagram.
 - Sequence Diagram
-
- The system likely follows these steps:
 1. **User:** Attributes - userID, name, email, password; Methods - login(), logout(), viewProfile()
 2. **Administrator:** Inherits from User; Methods - manageUsers(), viewReports()
 3. **Hostel Manager:** Inherits from User; Methods - allocateRoom(), manageMaintenance()
 4. **Warden:** Inherits from User; Methods - manageAttendance(), enforceRules()
 5. **Student:** Inherits from User; Attributes - studentID, roomNumber; Methods - submitMaintenanceRequest(), viewRoomDetails()
 6. **Maintenance Staff:** Inherits from User; Methods - updateMaintenanceStatus()
 7. **Room:** Attributes - roomID, status, capacity; Methods - allocate (), deallocate()
 8. **Maintenance Request:** Attributes - requestID, studentID, description, status; Methods - submitRequest(), updateStatus()
 9. **Report:** Attributes - reportID, type, content; Methods - generateReport()
 - Below Diagram show the UML diagram for the project named hostel management system integrated.

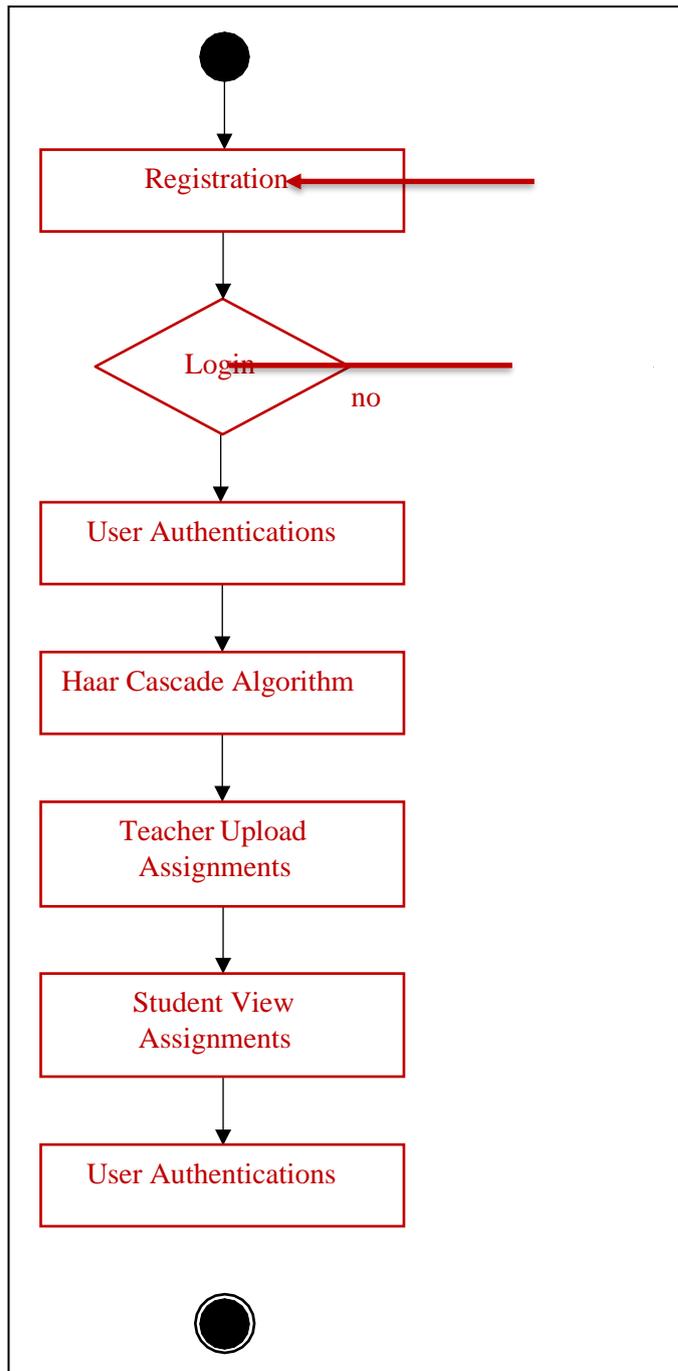


Fig. 6.3: Activity Diagram

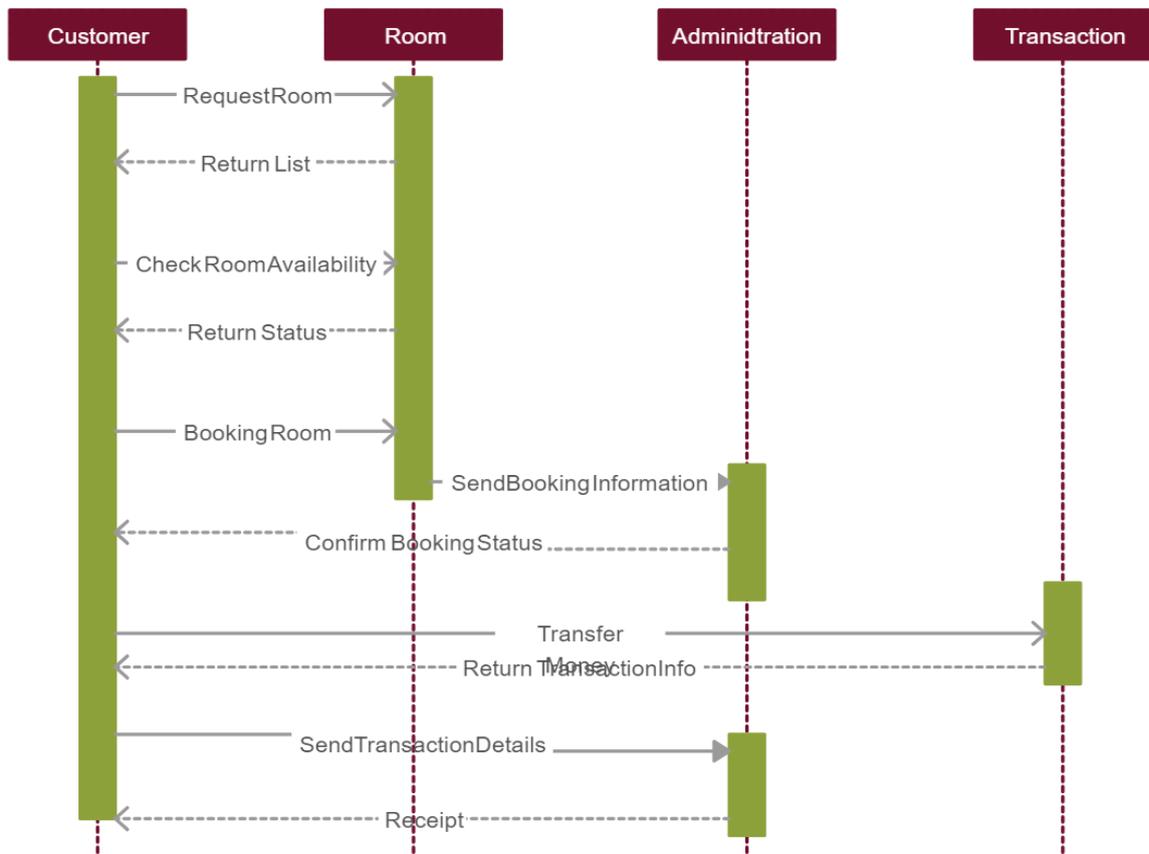


Fig 6.4: Sequence Diagram

Introduction:

This document analyzes the sequence diagram for a project involving secure facial recognition. This analysis aims to provide a clear understanding of the system's functionality and interactions between its components.

Sequence of Events:

- User Login
- User Action: The user (e.g., student, warden, hostel manager, administrator) opens the hostel management system application.
- System Response: The system displays the login screen.
- User Action: The user enters their username and password.
- System Response: The system validates the credentials.
- If valid, the system directs the user to the appropriate dashboard based on their role.
- If invalid, the system displays an error message.
- Dashboard Access
- System Response: After successful login, the system displays the dashboard with options relevant to the user's role (e.g., student dashboard, warden dashboard).
- Room Allocation (Hostel Manager)
- User Action: The hostel manager selects the room allocation option.
- System Response: The system displays a list of students and available rooms.
- User Action: The hostel manager selects a student and an available room.
- System Response: The system checks the availability of the room.
- If available, the system allocates the room to the student.
- If not available, the system notifies the hostel manager.
- System Response: The system updates the database with the new room allocation.
- System Response: The system sends a notification to the student about the room allocation.
- Submitting a Maintenance Request (Student)
- User Action: The student selects the maintenance request option.
- System Response: The system displays the maintenance request form.
- User Action: The student fills out the form with the details of the maintenance issue.

- System Response: The system records the maintenance request in the database.
- System Response: The system sends a notification to the maintenance staff.
- Maintenance Request Handling (Maintenance Staff)
- System Response: The maintenance staff receives a notification about a new maintenance request.
- User Action: The maintenance staff logs into the system and views the maintenance request details.
- User Action: The maintenance staff updates the status of the maintenance request (e.g., in progress, completed).
- System Response: The system updates the maintenance request status in the database.
- System Response: The system sends a notification to the student about the status update.
- Attendance Management (Warden)
- User Action: The warden selects the attendance management option.
- System Response: The system displays the list of students.
- User Action: The warden marks the attendance for each student (e.g., present, absent).
- System Response: The system records the attendance in the database.
- System Response: The system generates attendance reports if needed.
- Generating Reports (Administrator)
- User Action: The administrator selects the report generation option.
- System Response: The system displays options for different types of reports (e.g., room allocation, maintenance requests, attendance).
- User Action: The administrator selects the type of report to generate.
- System Response: The system fetches the necessary data from the database.
- System Response: The system generates the report and displays it to the administrator.

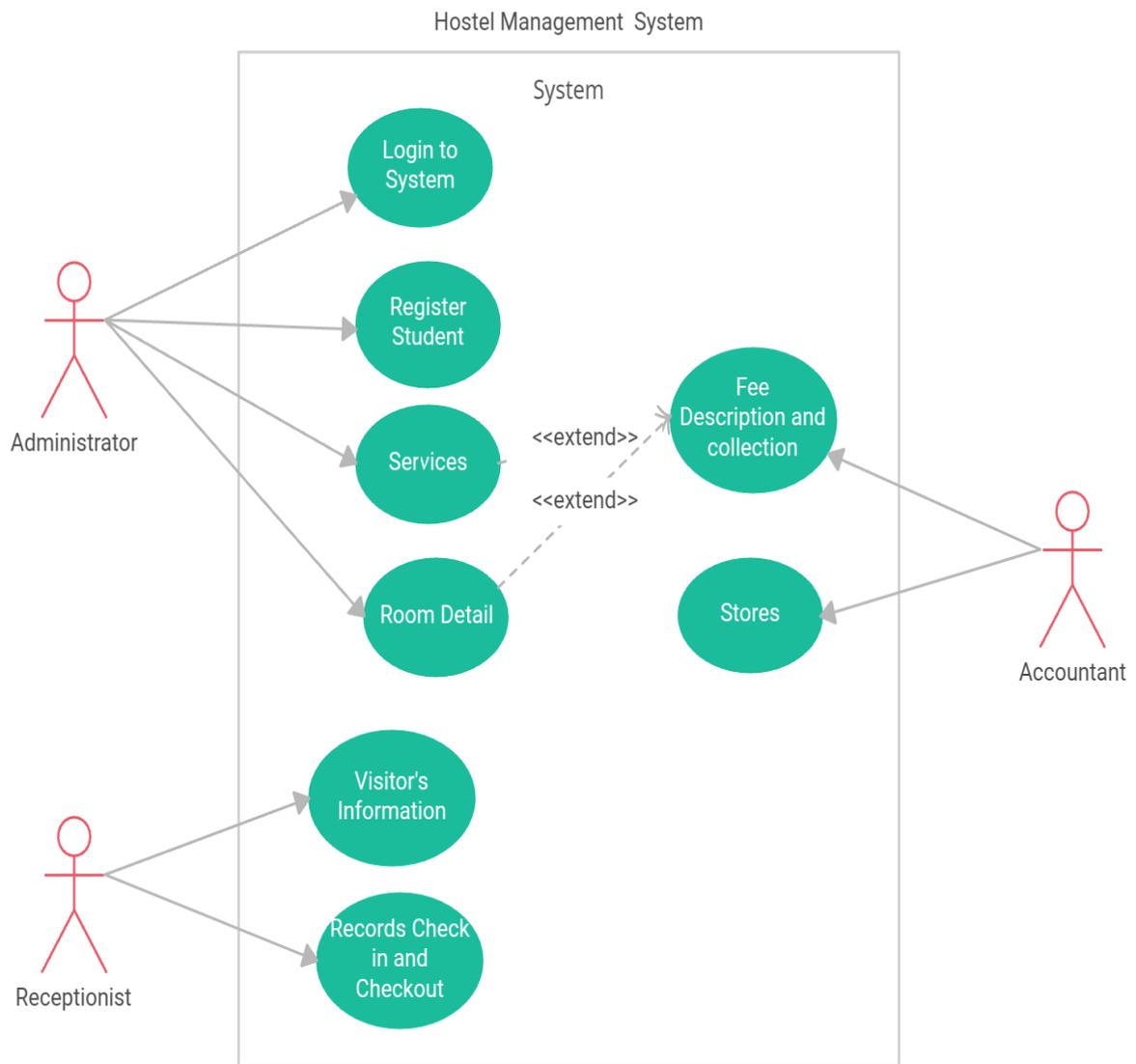


Fig 6.5: Use case Diagram

Actors:

- User: The individual attempting to gain access through face verification.

Use Case: Face Verification for HMS Access**Precondition:**

- The user has a valid account on the Hostel Management System (HMS).
- The user has a compatible camera connected to their device.

Basic Flow:

1. **Initiate Face Verification:** The user attempts to access a protected resource within the LMS that requires face verification.
2. **Activate Camera:** The LMS prompts the user to activate their webcam or camera.
3. **Face Detection:** The system processes each frame using image processing and face detection algorithms to identify the user's face.
4. **Decision Making:** Based on the results of face detection and liveness checks, the system makes a decision.
 - **Real Face:** If the system is confident, it's a live face, the HMS grants access to the user.
 - **Spoof Attempt:** If the system detects a spoof attempt, the HMS denies access and informs the user.
5. **Access Granted/Denied:** The HMS provides feedback to the user based on the decision (access granted, access denied due to spoof attempt).

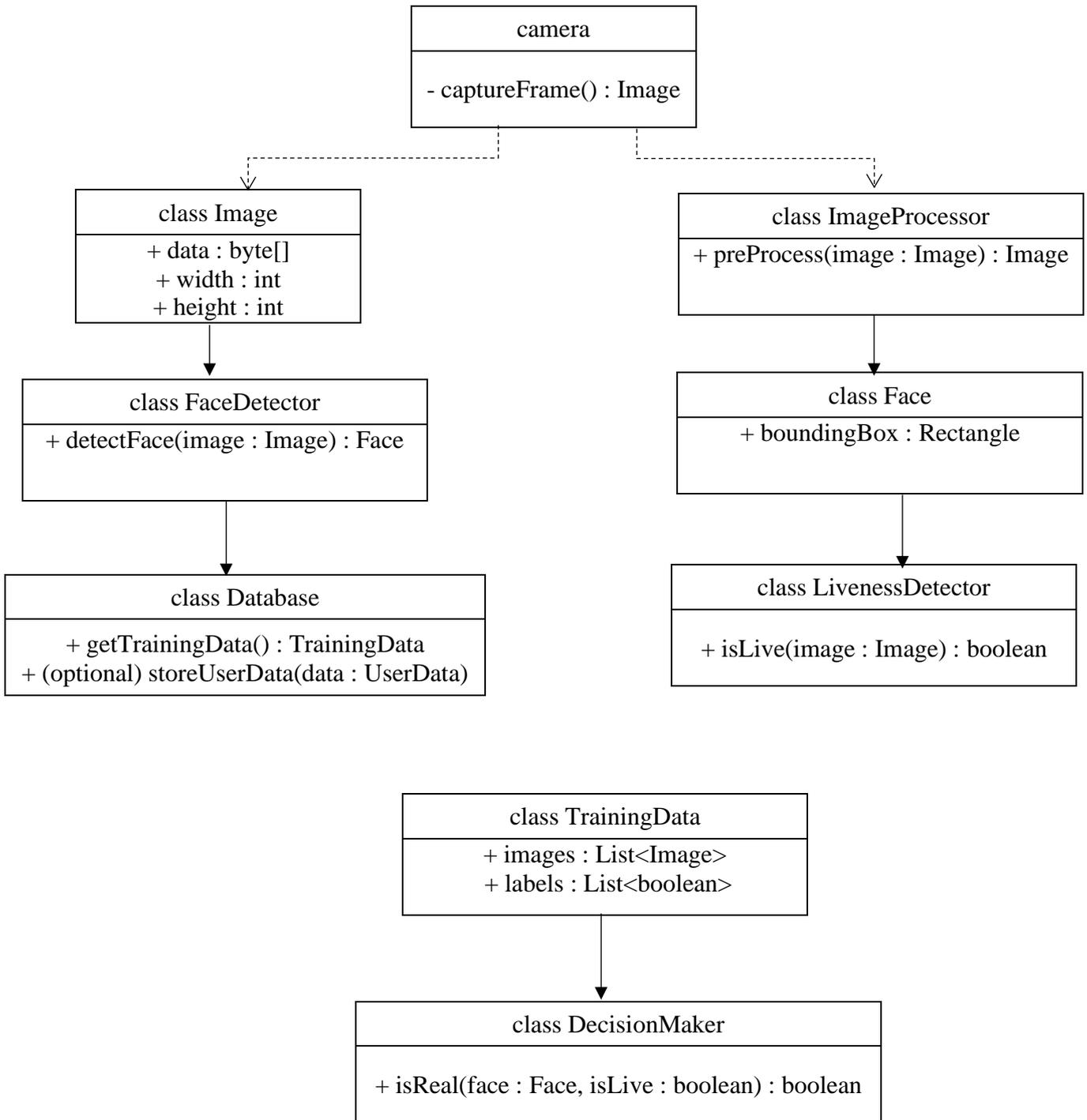


Fig 6.6: Class Diagram

Explanation:

- **Classes:** The diagram shows various classes representing the system components.
- **Attributes:** Each class has attributes that define its properties (e.g., Image has data, width, and height).
- **Operations:** Classes have methods (operations) that define their functionality (e.g., Camera.captureFrame() captures an image).
- **Relationships:** Arrows represent relationships between classes. Solid lines indicate usage (e.g., ImageProcessor uses Image objects). Dashed lines represent optional functionalities (e.g., storing user data in a database).
- **User Class (Example):** The User class is an example of how the system might interact with an HMS platform based on the decision. It's not part of the core system but demonstrates usage.

6.4 DESIGN CONSTRAINT

Here are some key design constraints to consider for our hostel management system demo:

Technical Constraints

- **Platform Compatibility:**
 - The system must be compatible with multiple platforms, such as web browsers (Chrome, Firefox, Safari) and mobile devices (iOS, Android).
- **Scalability:**
 - The system must be scalable to handle an increasing number of users, data, and transactions without significant performance degradation.
- **Performance:**
 - The system must have fast response times for user interactions and be capable of handling concurrent users efficiently.
- **Security:**
 - The system must ensure data security, including encryption of sensitive information (e.g.,

user credentials, personal data).

- Implement robust authentication and authorization mechanisms.
- Ensure compliance with relevant security standards and protocols.

- **Reliability and Availability:**
 - The system should have high availability with minimal downtime.
 - Implement redundancy and failover mechanisms to ensure reliability.

- **Integration:**
 - The system must support integration with third-party services, such as payment gateways, SMS/email notification services, and existing school/university systems.

- **Database Constraints:**
 - The database must handle large volumes of data efficiently.
 - Ensure data integrity and support for ACID (Atomicity, Consistency, Isolation, Durability) properties.

- **User Interface:**
 - The system must provide an intuitive and user-friendly interface for different user roles (e.g., students, wardens, hostel managers).

- **Maintenance:**
 - The system should be easy to maintain and update without requiring significant downtime.
 - Provide tools for regular backups and data recovery.

- **Localization and Language Support:**
 - The system must support multiple languages to cater to a diverse user base.

- **Reporting:**
 - The system should offer customizable and comprehensive reporting capabilities.
 - Legal and Regulatory Constraints

- **Data Privacy:**
 - The system must comply with data privacy regulations, such as GDPR (General Data Protection Regulation) and other local data protection laws.
 - Implement features for data anonymization and user consent management.

- **Accessibility:**
 - The system must comply with accessibility standards (e.g., WCAG - Web Content Accessibility Guidelines) to ensure it is usable by individuals with disabilities.
 - Financial Constraints

- **Budget:**
 - The development and operational costs of the system must stay within the allocated budget.
 - Consider cost-effective solutions for hosting, development tools, and third-party services.

- **Licensing:**
 - The system should use open-source or affordable licensed software components to minimize costs.

- **Resource Allocation:**
 - Ensure optimal use of resources, including development team members, hardware, and software.

- **Timeline:**
 - The system must be developed and deployed within a specified timeframe.

- **Stakeholder Requirements:**
 - The system must meet the specific needs and expectations of stakeholders, including students, wardens, and hostel management.

- **Customization and Flexibility:**
 - The system should be flexible enough to accommodate future enhancements and customizations as required by the hostel management.

6.5 SOFTWARE INTERFACE CONSTRAINT

1. User Interface (UI) Constraints

- **Consistency:**
 - The UI must have a consistent look and feel across all modules.
 - Use a uniform color scheme, typography, and layout structure.

- **Responsiveness:**

- The UI must be responsive and adapt to different screen sizes and devices (e.g., desktops, tablets, smartphones).

- **Accessibility:**

- The UI must comply with accessibility standards (e.g., WCAG) to ensure it is usable by individuals with disabilities.
- Include features like screen reader support, keyboard navigation, and high-contrast mode.

- **Intuitiveness:**

- The UI must be easy to navigate and understand, even for users with minimal technical expertise.
- Use clear labels, tooltips, and help documentation to guide users.

- **Performance:**

- The UI should load quickly and respond promptly to user interactions.
- Minimize the use of heavy graphics and optimize the loading of assets.

- **Feedback:**

- Provide immediate and clear feedback for user actions (e.g., form submissions, button clicks).
- Use visual indicators for ongoing processes (e.g., loading spinners).

- **Error Handling:**

- Display user-friendly error messages and guidance on how to correct mistakes.
- Prevent user errors with form validations and constraints.

2. Application Programming Interface (API) Constraints

- **Standardization:**

- Use standard protocols and formats (e.g., RESTful APIs, JSON) for communication between the front-end and back-end systems.

- **Security:**
 - Secure APIs with authentication and authorization mechanisms (e.g., OAuth, JWT).
 - Encrypt data transmitted between the client and server.
- **Documentation:**
 - Provide comprehensive and clear documentation for all APIs.
 - Include examples and usage guidelines for developers.
- **Versioning:**
 - Implement API versioning to manage updates and ensure backward compatibility.
 - Clearly indicate deprecated endpoints and provide migration paths.
- **Error Handling:**
 - Use standardized error codes and messages.
 - Provide detailed error information to help developers troubleshoot issues.

3. Integration Constraints

- **Compatibility:**
 - Ensure compatibility with existing systems and databases used by the hostel management (e.g., student information systems, financial systems).
- **Interoperability:**
 - Support data exchange with other systems through standard formats (e.g., CSV, XML) and protocols (e.g., SOAP, REST).
- **Data Synchronization:**
 - Implement mechanisms for data synchronization and consistency across integrated systems.
 - Handle conflicts and ensure data integrity during synchronization.

4. User Experience (UX) Constraints

- **Personalization:**
 - Provide personalized experiences based on user roles (e.g., student, warden, hostel manager).

- Allow users to customize their dashboards and preferences.

- **Simplicity:**
 - Simplify complex processes with step-by-step guidance and wizards.
 - Minimize the number of steps required to complete common tasks.

- **Localization:**
 - Support multiple languages and regional settings to cater to a diverse user base.
 - Ensure proper localization of content, including date formats, currencies, and measurements.

CHAPTER 7
DETAILED DESIGN DOCUMENT USING APPENDIX A AND B

7.1 ARCHITECTURAL DESIGN

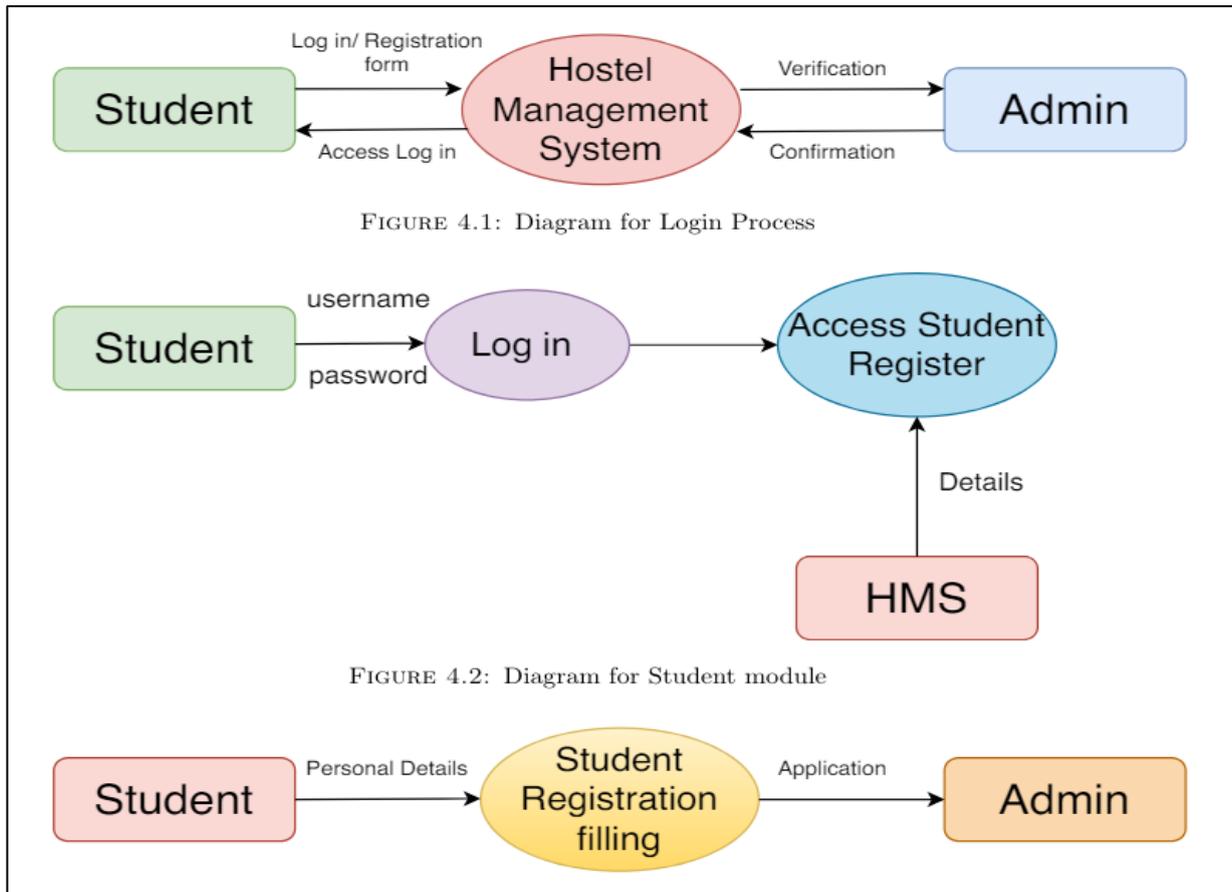


Figure 7.1: Architectural Design

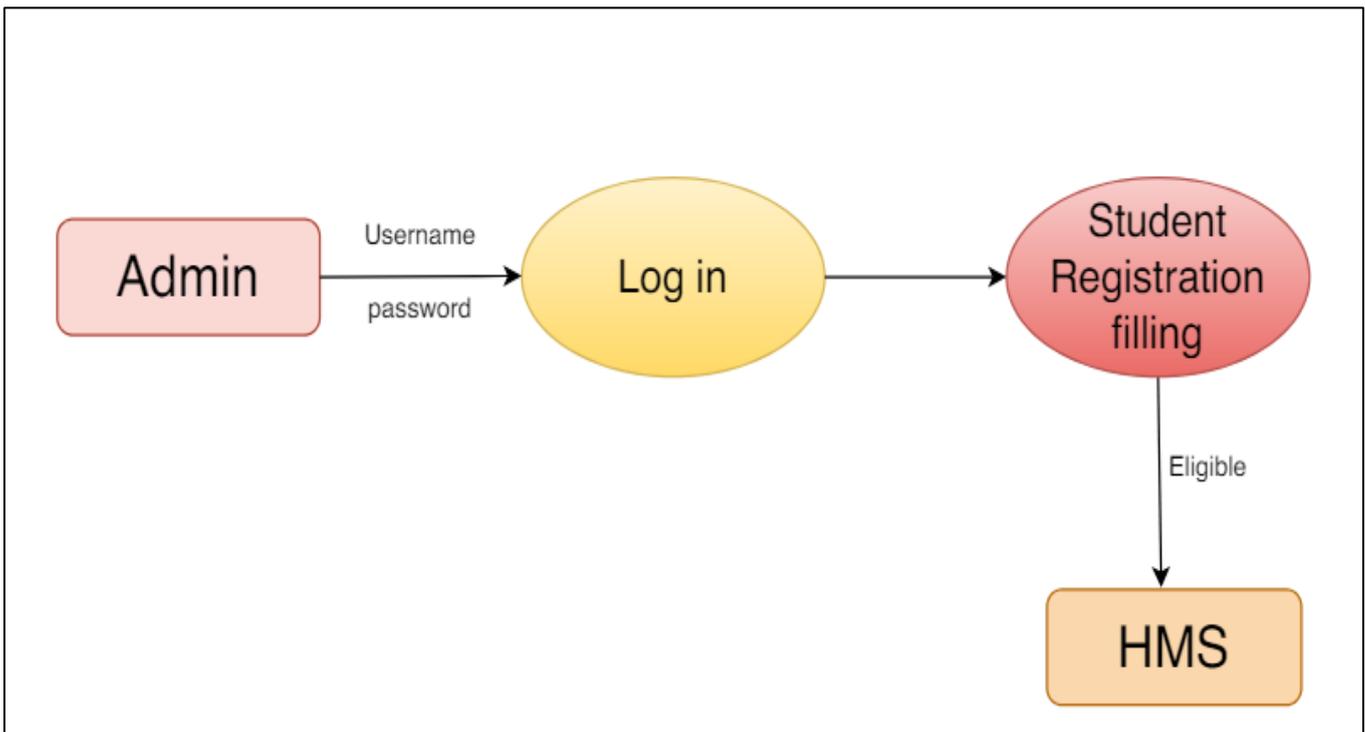


Figure 7.2: System Architectural Design

The system architecture you've outlined describes the key components involved in our hostel management system demo with the following stages:

High-Level Overview

The hostel management system can be designed using a multi-tier architecture, typically involving the following layers:

- *Presentation Layer (Front-End)*
- *Application Layer (Business Logic)*
- *Data Layer (Database)*
- *Integration Layer (APIs and Services)*

Architectural Components

1. Presentation Layer (Front-End)

- Purpose: To interact with the end-users, providing an interface for students, wardens, hostel managers, and administrators.
- Technologies: HTML, CSS, JavaScript, React.js/Angular/Vue.js (for web applications), React Native/Flutter (for mobile applications).
- Components:
 - User Interface (UI): Forms, dashboards, notifications, reports, etc.
 - User Authentication: Login, registration, password management.

2. Application Layer (Business Logic)

- Purpose: To handle the core functionality and business logic of the system.
- Technologies: Java/Spring Boot, .NET, Node.js, Python/Django or Flask.
- Components:
 - Room Management: Allocation, deallocation, room availability.
 - Student Management: Enrollment, profiles, attendance.
 - Reporting: Generating and managing various reports.

3. Data Layer (Database)

- Purpose: To store and manage the system's data.
- Technologies: MySQL, PostgreSQL, MongoDB, Oracle.

- Components:
- Database Schema: Tables for users, rooms, maintenance requests, attendance records, etc.
- Data Access Layer (DAL): Interfaces for querying and updating the database.
- Backup and Recovery: Regular data backups and mechanisms for recovery.

4. Integration Layer (APIs and Services)

- Purpose: To facilitate communication between the system and external services or other internal systems.
- Technologies: RESTful APIs, GraphQL, SOAP, OAuth for authentication.
- Components:
- API Gateway: Centralized entry point for API requests.
- Third-Party Integrations: Payment gateways, SMS/email services.
- Internal Services: Microservices handling specific functionalities (e.g., notification service).
- Description of Key Interactions
- User Interactions:
- Users interact with the system via the presentation layer using web or mobile applications.
- The front-end sends requests to the application layer via HTTP/HTTPS.
- Business Logic Execution:
- The application layer processes the requests, executing the core business logic.
- It interacts with the data layer to fetch or update data as needed.
- Data Storage and Management:
- The data layer handles all database interactions, ensuring data integrity and security.
- The DAL abstracts the database operations from the application layer.
- Notifications and Reporting:
- The system sends notifications (e.g., room allocation, maintenance updates) via integrated services.
- The reporting module generates various reports, pulling data from the database as required.

7.2 DATA DESIGN

Entities:

- **Hostel:**

Attributes:

- Hostel-id (primary key)
- name
- address
- description
- capacity (total number of residents)

- **Room:**

Attributes:

- room_id (primary key)
- hostel_id (foreign key referencing Hostel)
- room_type (e.g., single, double, dormitory)
- capacity (number of occupants)

- **Student:**

Attributes:

- student_id (primary key)
- name
- email
- phone_number
- check_in_date
- room_id (foreign key referencing Room)

- **Attendance:**

Attributes:

- attendance_id (primary key)
- resident_id (foreign key referencing Resident)
- date
- time
- image_path (path to the captured image for attendance verification)
- status (present, absent, unidentified)

- **Camera:**

Attributes:

- camera_id (primary key)
- location (e.g., main entrance, specific floor)
- status (active, inactive)

7.2.1 Internal Software Data

- **Image Storage:** Decide on a strategy for storing captured images. Options include storing them on the local server, a cloud storage service, or deleting them after a certain period for privacy reasons.
- **Facial Recognition (Optional):** If using facial recognition for attendance verification, additional data fields might be needed to store facial features or reference images for each resident.
- **Security:** Implement robust security measures to protect resident data, including access controls and encryption of sensitive information.
- **Data Retention:** Define policies for how long attendance data and images will be stored before deletion or anonymization.
- **Benefits of Camera Attendance:**
 - Improved Accuracy: Reduces manual errors associated with attendance tracking.

- Reduced Workload: Automates attendance recording, freeing staff for other tasks.
- Enhanced Security: Provides a visual record of who entered or exited the hostel.
- Time Efficiency: Enables real-time attendance monitoring.

- **Challenges of Camera Attendance:**

- Privacy Concerns: Implement measures to address resident concerns about data collection and storage of images.
- Lighting Conditions: Ensure proper lighting to optimize facial recognition accuracy.
- Cost: Requires investment in camera equipment and potentially facial recognition software.

7.3 COMPONENT DESIGN

Here's a breakdown of the key components we consider for our hostel management system demo project:

- **User Management:**

- User Registration: Allows students, staff, and administrators to register accounts in the system.
- Authentication: Provides secure login mechanisms for users to access the system.
- Role-based Access Control: Assigns roles (e.g., student, warden, admin) to users and controls their access to system functionalities.

- **Room Management:**

- Room Allocation: Assigns students to rooms based on availability and preferences.
- Room Details: Stores information about room numbers, capacities, and current occupants.
- Warden Assignment: Associates wardens with specific hostel blocks or floors for supervision.

- **Camera Management:**

- Camera Setup: Configures cameras at strategic locations within the hostel premises.
- Camera Integration: Integrates cameras with the system to capture attendance data.
- Camera Calibration: Adjusts camera settings for optimal performance and accuracy.

- **Attendance Management:**

- Attendance Recording: Captures student attendance using camera footage.
- Attendance Processing: Analyzes camera data to identify student presence and record check-in/check-out times.
- Real-time Monitoring: Provides live monitoring of attendance events for immediate intervention if needed.

- **Notification System:**

- Attendance Alerts: Notifies staff and administrators of irregular attendance patterns or anomalies.
- Attendance Updates: Sends notifications to students regarding their attendance status and reminders for regular check-ins.

- **Reporting and Analytics:**

- Attendance Reports: Generates reports on student attendance trends, including daily, weekly, and monthly summaries.
-
- Analytics Dashboard: Provides visual analytics tools to track attendance metrics and identify patterns or trends.

- **Integration with User Profiles:**

- User Matching: Matches camera attendance data with user profiles to accurately identify students.
- Profile Updates: Automatically updates student profiles with attendance records.

- **Maintenance and Support:**

- System Maintenance: Ensures the proper functioning of camera systems and attendance tracking mechanisms.
- Technical Support: Provides assistance to users experiencing issues with camera attendance or system functionalities.

- **Security Measures:**

- Data Encryption: Secures attendance data during transmission and storage to prevent unauthorized access.

- Access Control: Limits access to attendance data to authorized personnel only.
- Privacy Compliance: Ensures compliance with data privacy regulations to protect student information captured through camera attendance.

- **Scalability and Performance Optimization:**
 - System Scalability: Designs the system to handle increasing numbers of students and attendance data without compromising performance.
 - Performance Optimization: Implements efficient algorithms for processing camera footage and attendance records to minimize processing time and resource usage.

- **User Interface:**
 - Attendance Interface: Provides an intuitive interface for users to view attendance records, check-in/out statuses, and receive notifications.
 - Administrative Dashboard: Offers administrators and staff a centralized dashboard for managing attendance data, generating reports, and monitoring system health.

CHAPTER 8
PROJECT IMPLEMENTATION

8.1 INTRODUCTION

Implementation of our hostel management system with camera attendance entails leveraging modern technology to automate attendance tracking processes, enhance data accuracy, and streamline administrative workflows. By harnessing the power of camera surveillance and intelligent analytics, educational institutions can achieve real-time monitoring of student presence, improve security measures, and optimize hostel operations.

This implementation aims to revolutionize hostel management practices by providing a comprehensive platform that seamlessly integrates camera-based attendance functionalities with core hostel management features. Through this system, administrators can effortlessly track student check-ins and check-outs, receive instant alerts for irregular attendance patterns, and generate insightful reports to inform decision-making processes.

8.2 TOOLS AND TECHNOLOGIES USED

Programming Languages:

- Python: Often used for backend development, data processing, and integration with computer vision libraries.
- JavaScript: Used for frontend development, implementing interactive user interfaces, and integrating with web-based applications.
- Java, C#, or C++: Alternative options for backend development and system-level programming.

Web Development Frameworks:

- Django: A Python web framework for building scalable and secure web applications with backend APIs.
- Flask: A lightweight Python web framework suitable for small to medium-sized projects and rapid prototyping.
- React.js, Angular, Vue.js: JavaScript frameworks for building dynamic and interactive frontend interfaces.

Database Management Systems (DBMS):

- MySQL, PostgreSQL: Open-source relational database management systems (RDBMS)

commonly used for storing structured data in hostel management systems.

- MongoDB: A NoSQL database suitable for storing unstructured or semi-structured data, such as camera attendance records.

Camera Hardware and Software:

- IP Cameras: Network-connected cameras capable of capturing high-resolution video footage and streaming it over the internet.
- OpenCV (Open-Source Computer Vision Library): A popular open-source computer vision library used for image and video processing tasks such as face recognition, object detection, and motion tracking.
- TensorFlow, PyTorch: Deep learning frameworks with pre-trained models for implementing advanced computer vision algorithms, including face recognition.

Web Technologies:

- HTML5, CSS3: Standard markup and styling languages used for building web interfaces.
- JavaScript Libraries and Frameworks (e.g., jQuery, Bootstrap): Enhance frontend development with pre-built components, animations, and responsive design.

8.3 METHODOLOGIES / ALGORITHMS USED

Iterative Development:

- Break down the development process into small, manageable iterations or sprints, typically lasting 1-4 weeks.
- Each iteration focuses on delivering a specific set of features or functionalities, allowing for frequent feedback and adjustments based on stakeholder input.

Continuous Feedback and Collaboration:

- Engage stakeholders, including hostel administrators, staff, and students, throughout the development process to gather feedback and refine requirements.
- Regularly demonstrate working prototypes or incremental releases to stakeholders to validate assumptions, gather feedback, and ensure alignment with user needs.

Adaptability to Changing Requirements:

- Embrace change and prioritize responding to evolving requirements over rigid adherence to initial project plans.
- Use techniques such as user stories, backlog grooming, and sprint planning meetings to prioritize and adjust project scope based on changing priorities and stakeholder feedback.

Cross-Functional Teams:

- Form cross-functional teams comprising developers, designers, testers, and domain experts to collaborate closely and deliver high-quality software.
- Encourage communication and collaboration among team members to foster a shared understanding of project goals and foster collective ownership of deliverables.

Incremental Delivery of Value:

- Prioritize the delivery of features that provide the most value to users and stakeholders, starting with core functionalities and progressively adding enhancements.
- Use techniques such as user story mapping and Minimum Viable Product (MVP) definition to prioritize features based on their impact and feasibility.

Continuous Integration and Testing:

- Implement automated testing practices, including unit tests, integration tests, and acceptance tests, to ensure the reliability and quality of the hostel management system.
- Integrate testing activities seamlessly into the development process, with each code change triggering automated tests to detect regressions and ensure system stability.

Regular Reflection and Improvement:

- Hold regular retrospective meetings at the end of each iteration to reflect on what went well, what could be improved, and actions to take in the next iteration.
- Encourage a culture of continuous improvement by addressing process bottlenecks, technical debt, and team dynamics to enhance productivity and effectiveness.

Transparent Communication and Documentation:

- Maintain open and transparent communication channels within the team and with stakeholders to ensure everyone is aligned on project goals, progress, and challenges.
- Document decisions, requirements, and project artifacts in a transparent and accessible manner, enabling collaboration and knowledge sharing among team members.

8.4 ALGORITHM

1. Start

2. Initialization:

- Initialize the camera system and configure camera settings such as resolution, frame rate, and field of view.
- Set up the database to store student information, room allocations, and attendance records.

3. Face Recognition:

- For each detected face, apply a face recognition algorithm (e.g., Eigenfaces, Local Binary Patterns) to match the detected face with known student identities stored in the system's database.
- If a match is found, record the student's presence and timestamp the attendance entry.

4. Attendance Recording:

- Log the attendance record in the system's database, associating it with the corresponding student ID, timestamp, and location (e.g., hostel block, floor, room).

5. Real-Time Monitoring:

Display real-time attendance updates on a monitoring dashboard accessible to hostel administrators and staff.

6. Data Processing and Optimization:

- Implement algorithms to preprocess camera footage, including noise reduction, image stabilization, and contrast enhancement, to improve the accuracy of face detection and recognition.

- Optimize the attendance tracking algorithm for real-time processing, minimizing latency and resource utilization.

7. Security and Privacy Measures:

- Ensure data security and privacy by encrypting attendance records, restricting access to sensitive information, and complying with privacy regulations (e.g., GDPR, HIPAA).
- Implement secure authentication mechanisms to prevent unauthorized access to the system's features and data.

8. Error Handling and Recovery:

- Handle errors and exceptions gracefully, including network failures, camera malfunctions, and algorithmic errors.
- Implement error recovery mechanisms to resume normal operation after encountering errors or failures.

9. Scalability and Performance Optimization:

- Design the system to scale gracefully with increasing numbers of students and camera installations.
- Implement parallel processing and distributed computing techniques to optimize performance and handle large volumes of camera footage and attendance data.

10. End

CHAPTER 9
SOFTWARE TESTING

9.1 TYPE OF TESTING USED

To ensure the effectiveness and reliability of our face liveness detection demo, we will employ a combination of testing strategies:

- **Unit Testing:** Individual software modules (e.g., face detection) will be rigorously tested with controlled input data to verify their functionality.
- **Integration Testing:** Once modules are individually tested, we will integrate them and test their interaction with each other and with the LMS platform.
- **User Acceptance Testing (UAT):** We will involve potential users (students and faculty) to test the demo in a simulated HMS environment and gather feedback on its usability and effectiveness.
- **Security Testing:** We will conduct security testing to identify and address potential vulnerabilities in the system, particularly regarding data privacy and user authentication.

9.2 TEST CASES AND RESULTS

Here are Face Detection Test Case:

Test Case: Verify that the system accurately detects faces in captured frames.

Expected Result: The system should detect faces within a reasonable time frame and with high accuracy, minimizing false positives and false negatives.

Face Recognition Test Case:

Test Case: Verify that the system correctly recognizes known students' faces and associates them with their respective identities.

Expected Result: The system should match detected faces with stored student profiles in the database, displaying the correct student names or IDs.

Attendance Recording Test Case:

Test Case: Confirm that the system accurately records student attendance based on detected faces.

Expected Result: The system should timestamp attendance entries for each recognized student, storing the data in the system's database for future reference.

Real-Time Monitoring Test Case:

Test Case: Ensure that real-time attendance updates are displayed accurately on the monitoring dashboard.

Expected Result: The monitoring dashboard should provide up-to-date attendance information, including the list of present students and any detected anomalies or irregularities.

Performance Test Case:

Test Case: Assess the system's performance under various load conditions, including peak attendance periods.

Expected Result: The system should maintain acceptable response times and handle concurrent attendance tracking requests without significant degradation in performance.

Security Test Case:

Test Case: Verify that the system adequately protects student privacy and prevents unauthorized access to attendance data.

Expected Result: Attendance data should be encrypted during transmission and storage, and access to sensitive information should be restricted to authorized users only.

End-to-End Test Case:

Test Case: Test the entire system workflow, from student check-in to attendance recording and monitoring.

Expected Result: The system should accurately track student attendance in real time, updating the monitoring dashboard and recording attendance data in the system's database without errors or discrepancies.

CHAPTER 10
RESULTS

10.1 SCREENSHOTS

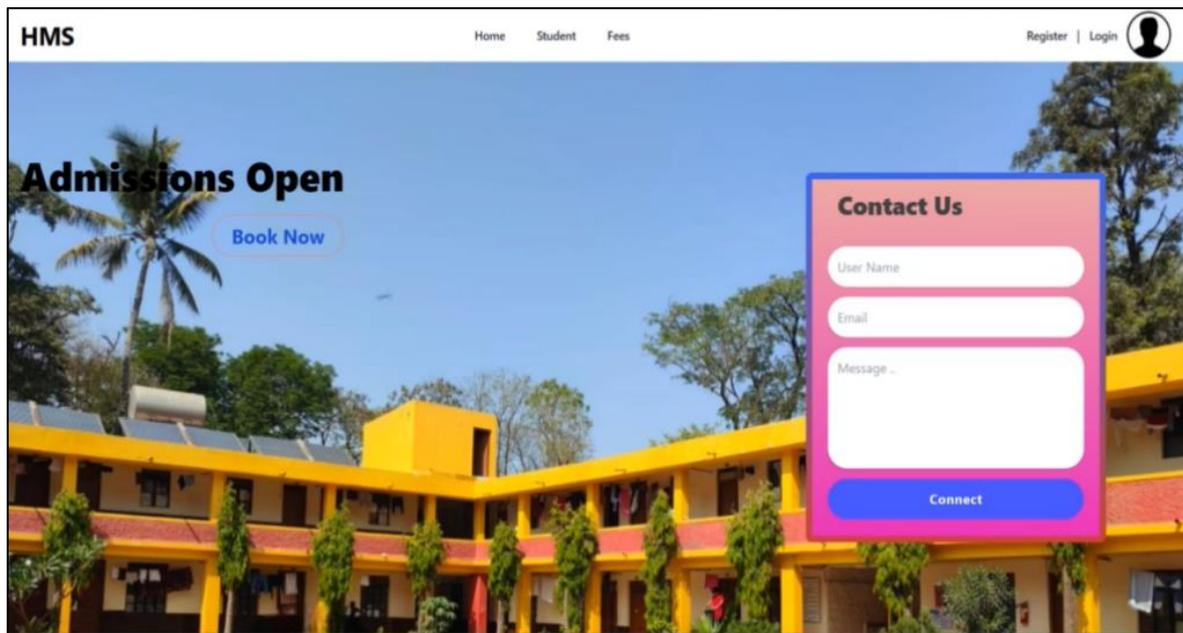


Fig 10.1: Home Page

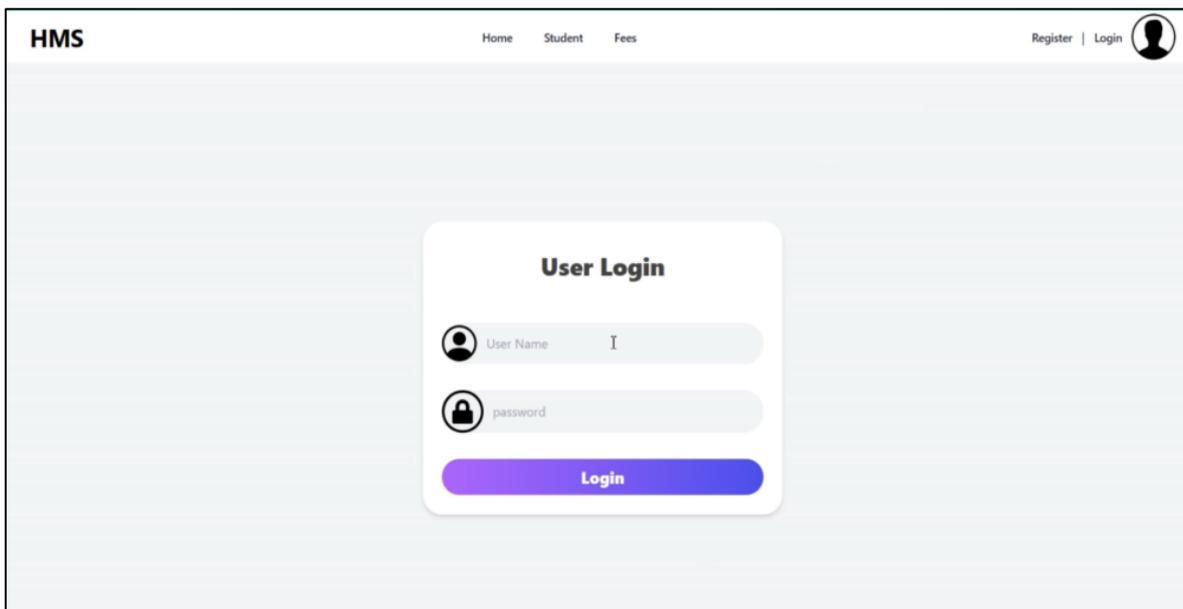


Fig 10.2: User Login Page

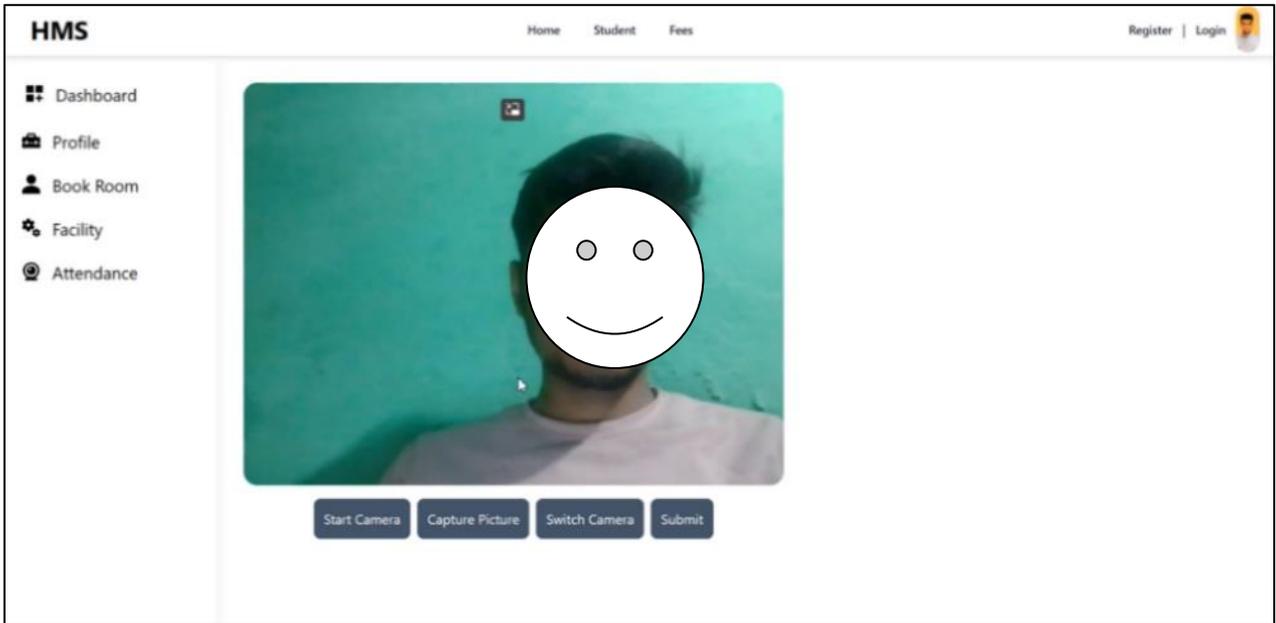


Fig 10.3: Camera Integration

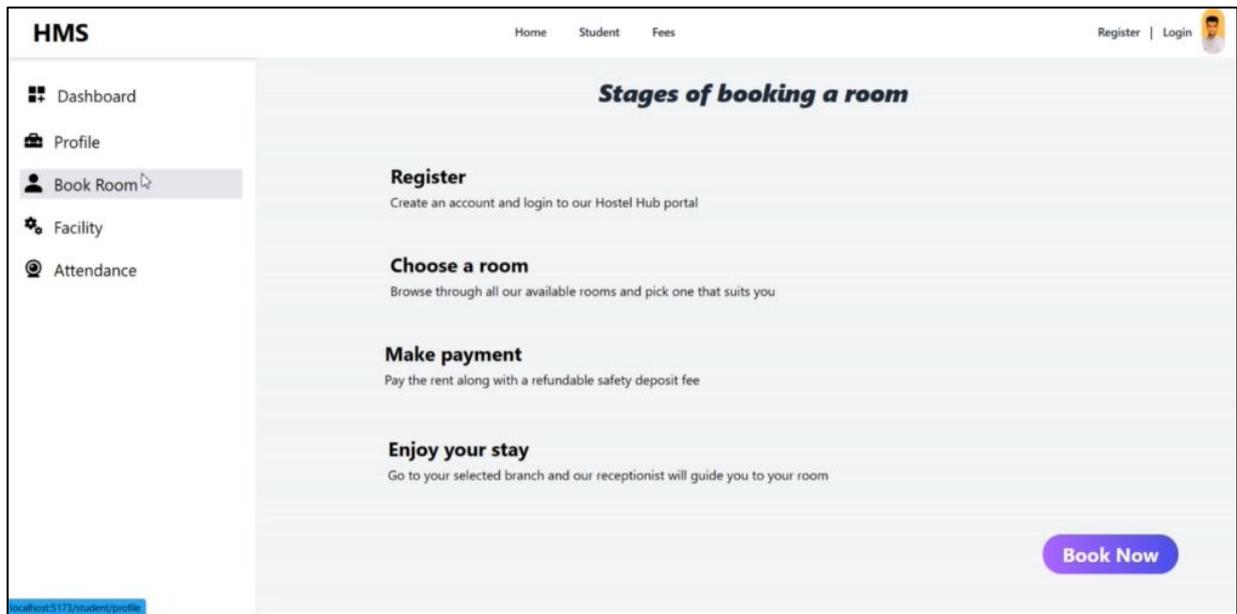


Fig 10.4: Room Booking

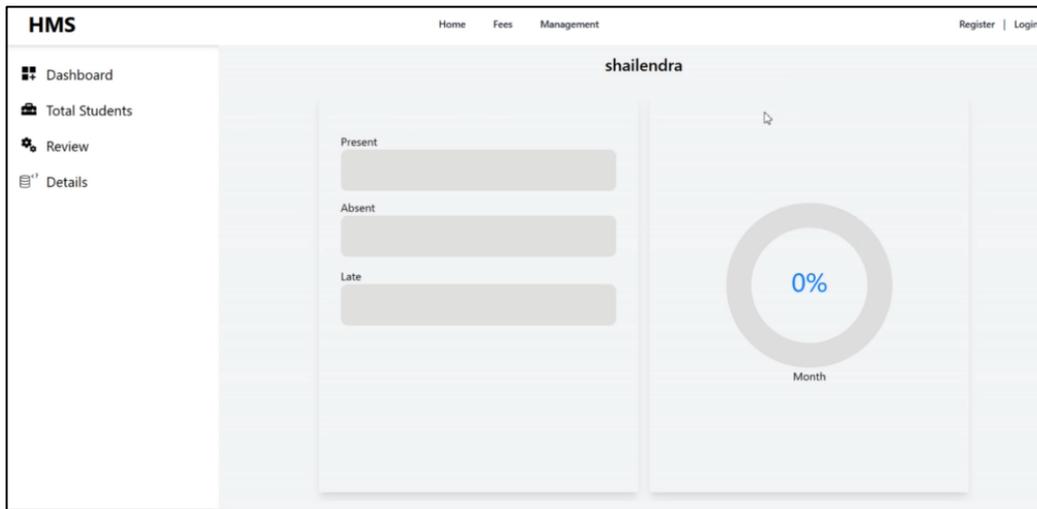


Fig 10.5: Student Attendance Details

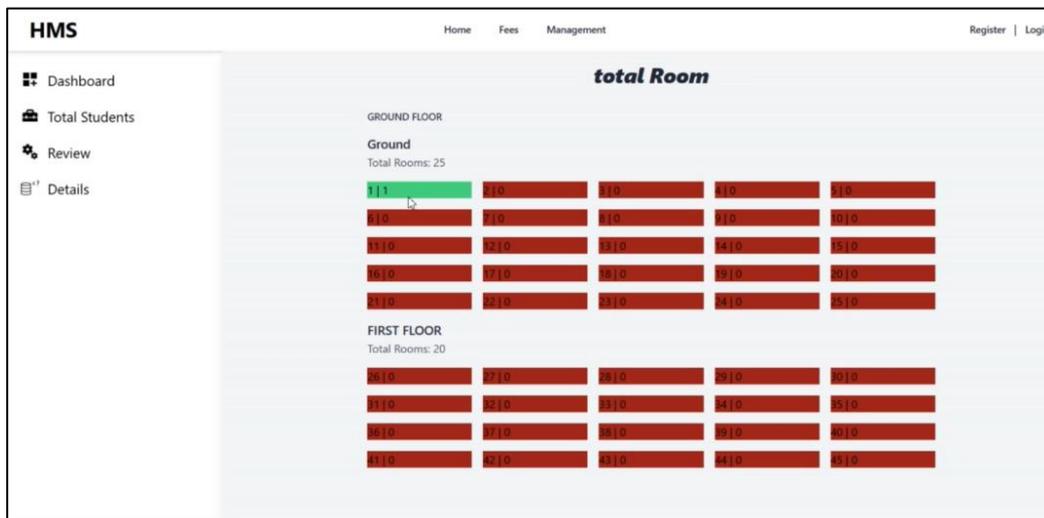


Fig 10.6: Total Rooms

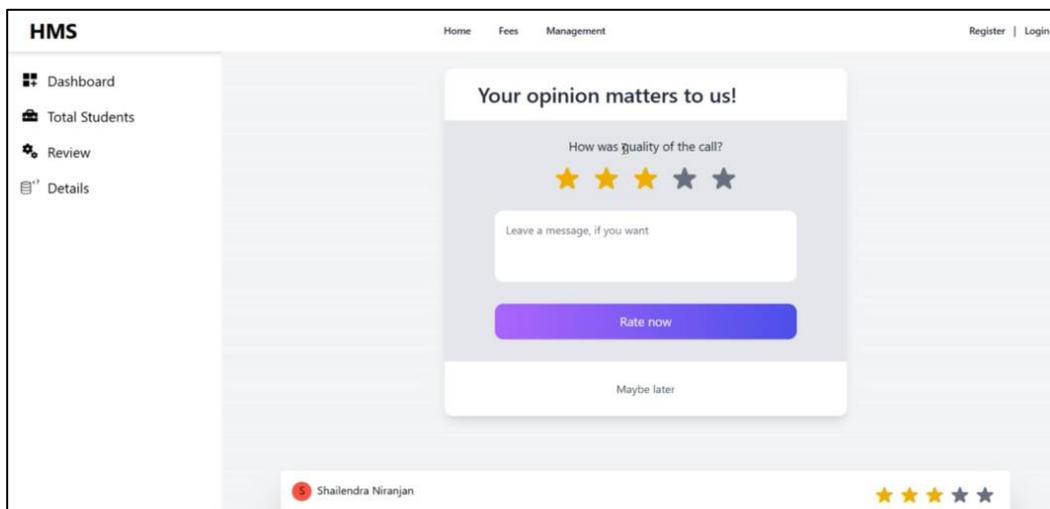


Fig 10.7: Review

CHAPTER 11
DEPLOYMENT AND MAINTAINANCE

11.1 INSTALLATION AND UNINSTALLATION

Anaconda:

1. Download the appropriate Anaconda installer for your operating system from <https://www.anaconda.com/download>.
2. Run the installer and follow the on-screen instructions. (Usually involves accepting license agreements and choosing installation paths)

Spyder (after Anaconda installation):

1. Open Anaconda Prompt (Windows) or Terminal (Mac/Linux).
2. Type `conda install spyder` and press Enter.

MySQL:

1. Download the MySQL installer for your operating system from <https://dev.mysql.com/downloads/mysql/>.
2. Run the installer and follow the on-screen instructions.

JAVA:

1. Download Java for Windows.
2. Run the Java installer
3. .Validate the JAVA_HOME setting.
4. Confirm the Java PATH variable was set properly.
5. Run a JDK command to verify Java install was a success.

Uninstallation:

Anaconda:

- **Windows:** Search for "Add or remove programs" and locate Anaconda. Click "Uninstall".
- **Mac:** Open Terminal and type `bash ~/anaconda/bin/uninstall.sh`. Press Enter and follow prompts.
- **Linux:** The uninstall process can vary depending on your distribution. Refer to the official Anaconda documentation for specific instructions.

Spyder (uninstalling Anaconda removes Spyder):

Follow the Anaconda uninstallation steps above.

MySQL:

- **Windows:** Search for "Add or remove programs" and locate MySQL. Click "Uninstall".
- **Mac:** Open Terminal and follow the official MySQL documentation for your specific version on how to uninstall (usually involves using a script or command-line tools).
- **Linux:** The uninstall process can vary depending on your distribution. Refer to the official MySQL documentation for specific instructions.

11.2 USER HELP

Understanding User Needs:

- **Target Audience:** Clearly define who your intended users are (students, instructors, administrators). Their technical literacy and familiarity with the system will influence the level of detail and terminology used.
- **Common Pain Points:** Identify the areas where users might encounter challenges or confusion. Gather feedback through surveys, user testing, or support tickets. Consider common roadblocks related to face verification, troubleshooting, or privacy concerns.
- **Information Hierarchy:** Prioritize content based on user needs. Start with the most frequently encountered issues and high-impact solutions.

Content Structure and Clarity:

- **Logical Organization:** Group related topics together for ease of navigation. Use clear headings, subheadings, and bullet points to break down information.
- **Concise Explanations:** Provide step-by-step instructions with screenshots or visuals where appropriate. Avoid overly technical jargon. Use active voice and simple language.
- **Contextualization:** Explain how procedures or concepts fit into the overall system usage.
- **Examples:** Illustrate concepts with real-world scenarios or use cases relevant to your users.

Accessibility and Usability:

- **Multiple Formats:** Consider offering the user help in different formats (e.g., online documentation, printable PDF, video tutorials) to cater to diverse learning styles and accessibility needs.
- **Search Functionality:** If the user help is extensive, implement a search bar to allow users to quickly find relevant information.
- **Consistent Terminology:** Use consistent terminology throughout the user help to avoid confusion. Consider a glossary for technical terms.
- **Mobile Compatibility:** Ensure the user help is responsive and accessible on mobile devices if applicable.

User Experience Focus:

- **Positive Tone:** Maintain a helpful and encouraging tone throughout the user help. Avoid patronizing language.
- **Error Handling:** Provide clear error messages and guidance on how to resolve common issues.
- **Human Touch:** Consider including screenshots or video tutorials showcasing successful interaction with the system to bolster user confidence.

Ongoing Improvement:

- **Feedback Mechanism:** Implement a feedback mechanism within the user help or system itself to allow users to report issues or suggest improvements.
- **Version Control:** Maintain a version history of the user help document to reflect changes, updates, or new features in the system.
- **Regular Updates:** Periodically review and update the user help based on user feedback, system changes, and best practices.

CHAPTER 12
CONCLUSION AND FUTURE SCOPE

SUMMARY

The hostel management system leveraging camera attendance revolutionizes attendance tracking by employing facial recognition technology, ensuring accurate and real-time monitoring of student presence. This innovative solution enhances security measures within hostel premises, automates administrative tasks, and offers comprehensive insights for hostel administrators.

By streamlining attendance recording processes and providing a centralized platform for monitoring and reporting, the system optimizes operational efficiency and resource utilization. Its scalable and adaptable design enables seamless integration with existing hostel management systems, contributing to a safer, more efficient hostel environment while mitigating manual errors and improving overall administrative effectiveness.

CONCLUSION

In conclusion, the hostel management system utilizing camera attendance presents a transformative solution to the traditional challenges of attendance tracking in hostel facilities. By harnessing facial recognition technology, it offers a seamless and accurate method for monitoring student presence in real time, enhancing security measures, and streamlining administrative processes.

This innovative system not only reduces manual effort and errors associated with traditional attendance methods but also provides valuable insights for hostel administrators through comprehensive monitoring and reporting capabilities. With its scalability and adaptability, the system not only meets the current needs of hostel management but also offers flexibility for future growth and advancements in hostel operations. Overall, the integration of camera attendance technology represents a significant step forward in optimizing hostel management practices, fostering a safer, more efficient, and data-driven environment for hostel administrators, staff, and students alike.

FUTURE SCOPE

More functionalities and address current limitations. Here's a glimpse into potential future advancements:

- **Advanced Image Recognition:**
 - Object Recognition: Identify items brought into or out of the hostel (luggage, valuables) for security purposes.
 - Activity Recognition: Analyze camera footage to detect suspicious behavior or potential safety hazards.
- **Integration with Smart Devices:**
 - Facial Recognition Door Locks: Cameras could unlock doors upon facial recognition, eliminating the need for physical keys.
 - Smart Room Automation: Camera data can trigger adjustments in room temperature, lighting, or other smart devices based on resident presence.
- **Enhanced Analytics and Reporting:**
 - Heatmap Generation: Analyze resident movement patterns within the hostel to optimize common area layouts or staff scheduling.
 - Predictive Maintenance: Facial recognition data combined with sensor data (temperature, humidity) could predict potential maintenance issues in rooms.
- **Privacy-Preserving Techniques:**
 - Data Anonymization: Anonymize attendance data after a certain period, protecting resident identities while retaining valuable insights.
 - Differential Privacy: Add noise to data to improve privacy while maintaining data utility for analysis.
- **Focus on Resident Well-being:**
 - Sleep Monitoring (Optional): Analyze sleep patterns (with resident consent) to identify potential sleep disturbances and offer resources for improved sleep hygiene.
 - Social Interaction Analysis (Optional): Anonymously analyze camera data to identify residents who might be isolated and offer opportunities for social interaction.

ANNEXTURE A
REFERENCES

References

1. Tanmay Nandanwar and Priyanka Bahu Tule -A STUDY ON SHIFT TOWARDS DIGITIZATION OF HOSTEL ROOM ALLOTMENT FOR A UNIVERSITY February 2020 [DOI:10.1109/ic-ETITE47903.2020.117]
2. Askar, Abhishek & Gaur, Saumya & Deolikar, Sejal & Ubale -DEVELOPMENT OF AN AUTOMATED HOSTEL FACILITY MANAGEMENT SYSTEM Journal of Science and Engineering [Vol. 5 (1), 2014, 01-10 - ISSN: 2331-5172]
3. Prof. Shyamsundar Magar, Ms. Sakshi Said and Mr. Rohit Jadhav -HOSTEL MANAGEMENT SYSTEM AND AGGREGATION ONLINE HOSTEL MANAGEMENT International Journal of Advanced Engineering & Science Research. 2021 JETIR October 2021, [Volume 8, Issue 10 ISSN-2349-5162]
4. Sreerambabu, Dr Riyaz, Mr Rajkumar (2023). GE HOSTEL INVENTORY MANAGEMENT SYSTEM COLLEGE
5. Narkhede, Prof & Bamgude, Rutuja & Sonawane, Mayuri & Shevade, Mandar- HOSTEL MANAGEMENT SYSTEM (HMS) – [Volume 10 Issue IV Apr 2022 ISSN: 2321-9653]
6. ISSN RAJKUMAR, G. & SUNDARI (2018). HOSTEL MANAGEMENT SYSTEM BASED ON FINGER PRINT AUTHENTICATION- [DOI:10.13005 -11.04.08]
7. Dinesh.B, Gogul Nithin.R, Pavatharani.R, Sneha.R, C.Senthilkumar (2022).AUTOMATION USING DESIGN THINKING THE RESEARCH OF DEVELOPMENT FRAMEWORK BASED ON MVC PATTERN [DOI:10.1007-13369-021-06087-4]
8. Wei Cui, Lin Huang and LiJing Liang (2018). FRAMEWORKS USABILITY IN WEB APPLICATION DEVELOPMENT [DOI:10. 35940.C1020.1083S19]
9. Mr. A.Aswar, Mr. A.Ganesan, Dr. V.Kavitha, Mr. V.Karthicksabri PG & Research Department of Computer Applications, on HOSTEL AUTOMATION SYSTEM. IJCSMC, [Vol. 8, Issue. 9, September 2019, pg.161 – 166 ISSN 2320–088X]
10. SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE) – [volume 3 Issue 4–April 2016 ISSN: 2348 – 8387]
11. Poojashree1, Ananth MURTHY ONLINE HOSTEL MANAGEMENT International Journal

of Advanced Engineering & Science Research (IJAES) [Volume 5, Issue 1, March 2017]

12. Kulasekhara S, HOSTEL MANAGEMENT SYSTEM USING SERVICE NOW European Journal of Molecular & Clinical Medicine, 2020, [Volume 7, Issue 4, Pages 1078-1084]
13. Robert S. Kaplan and David P. Norton Mastering the Management System [January 2008: 62–77.]
14. Lore: A DATABASE MANAGEMENT SYSTEM FOR SEMISTRUCTURED DATA Publication SIGMOD Record [DOI:10.1145/262762.262770]
15. Kesh Prasad, Philip W. Wirtz & Larry Yu MEASURING HOTEL GUEST SATISFACTION BY USING AN ONLINE QUALITY MANAGEMENT SYSTEM [DOI:10.1080/19368623.2013.805313]

ANNEXTURE B
COMPETITION CERTIFICATE



SAVITRIBAI PHULE PUNE UNIVERSITY

Internal Quality Assurance Cell (IQAC)

AAVISHKAR - 2023

ZONAL LEVEL RESEARCH PROJECT COMPETITION

Certificate of Participation

This is to certify that Mr./Miss. SUJAL THAKARE
of Anantnagar Pawan COE & Research College/Institute has participated
in Zonal Level Research for UG / PG / Post PG (M.Phil. / Ph.D.) at **"AAVISHKAR 2023"** Research
Festival, Under ENGINEERING & TECH category, held at AISSMS COE
PUNE during 27-10 2023.


Coordinator


Principal


Prof. Sanjay Dhole
Director, IQAC



SAVITRIBAI PHULE PUNE UNIVERSITY

Internal Quality Assurance Cell (IQAC)

AAVISHKAR - 2023

ZONAL LEVEL RESEARCH PROJECT COMPETITION

Certificate of Participation

This is to certify that Mr./Miss. RACHANA PATEL
of Anantnagar Pawan COE and Research College/Institute has participated
in Zonal Level Research for UG / PG / Post PG (M.Phil. / Ph.D.) at **"AAVISHKAR 2023"** Research
Festival, Under ENGINEERING & TECH category, held at AISSMS COE
PUNE during 27-10 2023.


Coordinator


Principal


Prof. Sanjay Dhole
Director, IQAC


Marathwada Mitra Mandal's College of Engineering
Accredited by NAAC with 'A++' grade | Best College Award by SPPU



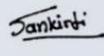
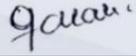
CICS
 CULTIVATING IDEAS CRAFTING SOLUTIONS

CERTIFICATE OF APPRECIATION

This certificate is presented to

Sujal Suresh Thakre

for participation in CICS IDEATHON conducted by FMCIII in collaboration with ISTE SC' AND ACM SC' at Marathwada Mitra Mandal College Of Engineering, Pune.

			
Dr.S.S.Shiravale <small>Secretary ISTE SC, MMCOE</small>	Mr.N.S.Dhavase <small>Associate Professor, MMCOE</small>	Dr.V.N.Gohokar <small>Principal, MMCOE</small>	Dr.C.R. Talathi <small>C.E.O, FMCIII</small>


Marathwada Mitra Mandal's College of Engineering
Accredited by NAAC with 'A++' grade | Best College Award by SPPU



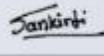
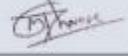
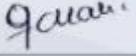
CICS
 CULTIVATING IDEAS CRAFTING SOLUTIONS

CERTIFICATE OF APPRECIATION

This certificate is presented to

Sadaf Azam Shaikh

for participation in CICS IDEATHON conducted by FMCIII in collaboration with ISTE SC' AND ACM SC' at Marathwada Mitra Mandal College Of Engineering, Pune.

			
Dr.S.S.Shiravale <small>Secretary ISTE SC, MMCOE</small>	Mr.N.S.Dhavase <small>Associate Professor, MMCOE</small>	Dr.V.N.Gohokar <small>Principal, MMCOE</small>	Dr.C.R. Talathi <small>C.E.O, FMCIII</small>



ANNEXTURE C
PAPERS SUBMITTED



Hostel Management System

Rachana Rajendra Patil¹, Aditya Krishna Sapkal², Sujal Suresh Thakre³, Sadaf Azam Shaikh⁴, Prof. Rama Gaikwad⁵

1,2,3,4,5BE Student, Department of Computer Engineering ABMSP's Anantrao Pawar College of Engineering and Research Pune, India

⁵Assistant Prof. Department of Computer Engineering ABMSP's Anantrao Pawar College of Engineering and Research Pune, India

ABSTRACT

The Hostel Management System is an application which has been designed for the management of different activities within a hostel. The number of colleges and universities is growing rapidly over the past few years. Consequently, the number of accommodation rooms is also increasing "to provide housing for students" at this institution. Therefore, it is a lot of work for the person in charge of the lodging facility and software is usually not available to do so. The project aims to address the challenges of running a hostel by preventing issues arising from manual operations. Identifying weaknesses of existing systems leads to the design of a Computer System that will be compatible with an existing system, making it more user friendly and further website oriented. The effectiveness of the system could be improved, thereby overcoming its negative effects.

Keywords: Online Hostel Management, Web Application, Database Management.

1. INTRODUCTION

Using a manual system is not advised in the modern era of automated systems, whether they be hardware or software. Typically, hostels lacking a management system are closed once a year. Manual verification of the registration forms and other data recovery procedures is carried out, in many cases, by handwriting on paper. Therefore, it is possible to prevent a recurrence. using an automatic mechanism. A computerized system that will assist in reducing the amount of human input was designed in response to the shortcomings of the current methods. With this system in place, by improving the efficiency of the system, it is possible to improve on the weaknesses that are present under the current manual method. It is intended, with the hostel administration in mind, to allow student records of their rooms and belongings to be preserved. It's going to save them from the heavy manual work where it would be very difficult to find student records, college invoices, or any information on those who left the dorm in the first place.

A. Objectives

Managing the specifics of rent, allotments, hostels, rooms, and payments is the major goal of the hostel administration system. It keeps track of all the data pertaining to rent, beds, payments, and rent. Access is restricted to the administrator only because the project was created with administration in mind. The project's goal is to develop an application that will lessen the amount of human labor required to manage the hostel, beds, rent, and allotments. It keeps track of every piece of data regarding payments, rooms, and hostels.

- The platform provides search functions based on several criteria. Payments, Rooms, Hostel, and Rent
- Personnel data for students, employees, and courses is also sold online via the College Management System.
- Offer the ability to make reservations.
- You may make updates to your website without hiring a web designer.
- Maintains a record of all the details regarding beds, rooms, allotments, etc.
- Oversee the allocation of data.
- Displays details and an explanation of the rent and hostel
- To improve the effectiveness of rent and allocation management.
- It deals with keeping an eye on room transactions and information.
- Keep track of the rental details

II. PROBLEM DEFININATION

The maintenance and operation of a hostel bring with them many disadvantages. Particularly when it comes to the manual system. As the majority of hostels are run by a single manager, the officer may not even know how many students there are in his room. To find out if a room is inhabited or not, he must proceed room by room. People may occasionally owe money at the hostel; these debts are recorded on papers, large notebooks, and occasionally receipts. It would never be possible to determine whether a student is due money or not if the books disappear or are stolen. It also becomes challenging to assign rooms since the officer might not know which rooms are accessible. Finding all the stories at some of these hostels that have a large number of rooms would be more time-consuming, thus making it hard to find a room. Furthermore, it's possible that the officer is unaware of how many pupils are in a room or whether it's full.

A. Existing System

The current approach requires a lot of work, is manual, and takes a long time. While we may apply online for hostels under the current system, the allocation procedures are still done by hand. Corruption in the process of allocating resources and calculating hostel fees might result from it. Complaint registration and mess calculation are not handled by the current system.

B. Proposed System

The goal of this project is to provide a system for recording and displaying data about or within a hostel. The hostel officer will be able to oversee the hostel's operations with the use of this technology. This system would offer comprehensive details on each hostel student. It will show whether rooms are available or not and the number of people in a particular room. Information on students who have made complete payments or who still owe will also be provided via this. Additionally, a report on the summary information about the fees and invoices that students are due will be given via this system. There is also a user module available for the hostel officer or personnel. Additionally, the administrator will have access to an administrator module where they may add, update, and remove personnel data. Using PHP and XAMPP servers, this system will be constructed using the Software Development Life Cycle (SDLC). While XAMPP is appropriate for databases due to its extensive capabilities and security, PHP and JS are better suited for creating and designing web-based applications

III. LITERATURE REIVIEW

In [1], a hostel's web version was suggested. Management system and integration of an RFID system for gate pass purposes. This web interface was created using PHP-MySQL technology. It reduces the efforts made by the hostel's manager and owner while managing the hostel. It removes the pen and paper idea, which we have been using since ancient times. This system has an administrator, warden, and student modules, each with their own different dashboards and roles. In addition to providing solutions for manual hostel management problems, the framework that was developed also provides data, such as hostel, student, and room data. The administrators of this system are able to send notifications or notices to students through the email system. It is a secure online website accessible anywhere on the globe. They proposed the use of RFID tags for the tracking purposes of students. RFID tags are given to students during hostel registration, so administrators can track student activity on a single dashboard. In [2], a web-based hostel management system is proposed. The three main elements are room assignment, automated bill computation, student information, automatic bill calculation, and room allocation. Students and administrators have different dashboards. To take a seat in the hostel, the student must register his or her details on the portal. The administrator will allot the room number along with assets like chairs, tables, and beds. In [3], a web-based hostel management system is proposed. This web portal is developed using ASP.NET and SQL server technology. This system has three actors: the administrator, the hostel clerk, and the student. This system provides facilities like student-driven hostel registration, invoice generation and printing, automatic due calculation and notification, student hostel clearance management, etc. The online gate pass application form for hostel students has been proposed in [4]. Gate Pass Management System is the software application used to manage the entry and exit of students on short-term and longterm leave. The hostel warden issues the gate pass after receiving approval from the HOD, hostel warden, or hostel manager. Or hostel manager and issued by the hostel warden. In this system, the student makes a request for entry or leave by filling in the different details. The higher authority verifies the details from the database; if they are valid, permission is granted or denied. The respected student is notified by email about his or her confirmation before time. The system's purpose is to document the pupils' actions and specifics. The system's goal is to "save paper" by utilizing technology. In [7], the personality-based hostel allocation system is shown by the fact that the pupils are assigned to rooms based on their temperaments. They use a first-come, first-served approach for student registration, where the registration system will ask a few questions and the student has to answer them. Their answers are then recorded in a database through a web portal. The approach to student selection is automatic and adopted from Eric Jorgenson's open-four temperament scale test, which is used to develop a sorting algorithm that judiciously detects and allocates students to available rooms within hostel accommodation. This system is web-based and made up of PHP and MySQL technology. This system consists of the implementation of the above-described algorithm, which processes the personality based on the answers given by students to the questions, sorts the students according to their personalities, and allocates the room only. The allocation of bed space and any other hostel amenities are excluded using this method. In [5], a desktop-based hostel with just one user management software is demonstrated. It is made up using Microsoft's Visual Basic, and it consumes SQL Server as the backend. It has only one user, known as an administrator. It performs student management, maintains IN-OUT records, hostel management, and report generation. In [6], a biometric authentication-based hostel management system is demonstrated. One of the most common problems at the hostel is security. To keep up with day-to-day attendance verification is a sophisticated and time-consuming system for hostel management. Wardens manually track hostel students' attendance under the current method. This study addresses the prevention of a complete issue in the hostel management system and also suggests a monitoring mechanism. This system has only one user as

administrator, which can be a college principal or warden. The dormitory uses a biometric method to accommodate a large number of students. This technology automatically tracks students' arrivals and departures from the dorm and sends parents alert SMS messages to ensure their safety.

IV. REQUIREMENT ANALYSIS

A. Functional Requirements

1. The user shall generate a user profile containing the following information: user account number, full name, address, phone number, and room number.
2. Users will change the status of their dues in the database according to whether they are paid or not.
3. Users must allow the warden to add new users to the system's database.
4. The user must allow the guardian to cancel the registration from the system's database.

B. Non-Functional Requirements

1. The system shall be capable of operating reliably. It's always got to be raised up and running.
2. A high performance should be achieved by the system.

C. Hardware Requirements

A major task related to software development is the section on hardware configuration. The speed and efficiency of the whole system may be negatively affected by a lack of random-access memory. This process must be very strong enough to run all the operations. You should be able to store a program or file on your hard drive. A. Size of cache: 512 KB. B. RAM: 512 MB (minimum). C. Network Card: A 100 Mbps speed may be achieved with any card. Network connection: UTP, or coaxial cable connection. Printer: An inkjet or laser-color printer provides at least 1000 dpi. Hard disk: 80 GB. E. Mouse: 104 keys (US Key, Serial, USB, or PS/2).

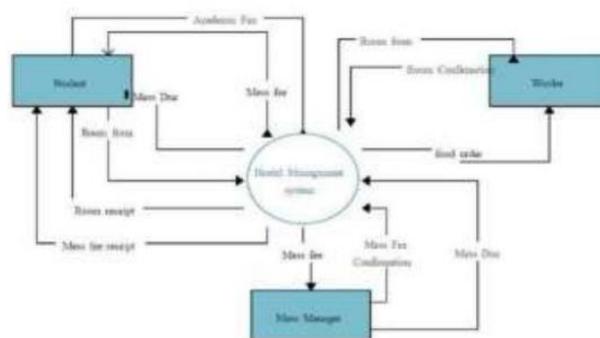
D. Software Configuration

The compatibility section of the software is one of the most important parts of a system development, since it undergoes geometric evolution on the market. The chosen software should be compatible with the firm, as well as to one user, and it should be viable for the system. This document provides a comprehensive explanation of the software requirement specification. The study of requirement specification is focused especially on the functioning of the system. It enables the analyst or developer to comprehend the system, the task at hand, the desired performance level, and the associated interfaces that need to be set up.

1. Language to use: latest version of HTML and PHP, JavaScript, AJAX, CSS and Apache Server
2. Database: MySQL XAMPP server
3. User Interface: HTML, AJAX, CSS, JFRAME
4. Web browser: Internet Explorer 8 or later, Mozilla and etc. Chrome
5. Software: MySQL Server and XAMPP 6. Operating system: Windows XP or newer versions

V. SYSTEM DESIGN

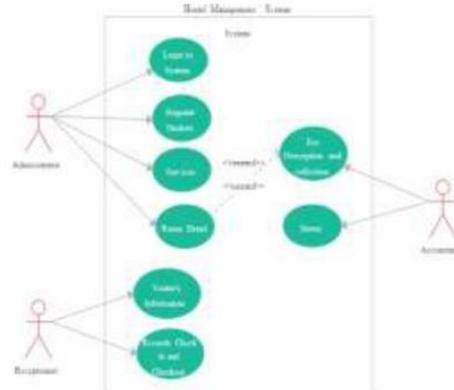
A. Data Flow Diagram



A data flow diagram for a hostel management system is a visual representation that illustrates the flow of information within the system. It highlights the data movement between them by identifying the processes, data sources, and destinations. A DFD would highlight activities related to hostel

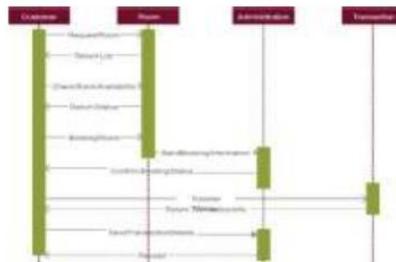
administration, such as student check-ins, room allocations, fee payments, and record-keeping. Each of these tasks is represented as a process node, and the direction of data flow is indicated by arrows. This graphical tool helps with the analysis, planning, and optimization of the system's activities for increased efficacy and efficiency by offering a simple, clear perspective.

B. Use case Diagram



The use case diagram is a graphic depiction of the interaction among the elements of the hostel management system. The hostel management system's system needs are identified, outlined, and organized using this process. In the context of the Hostel Management System, there are four key actors: the super admin, system user, allots, and hostel owner. They are responsible for carrying out various use cases, including managing the hostel, beds, rooms, rent, allots, payments, hostel facilities, users, and preparing the full hostel management system. The Hostel Management System's UML use case diagram displays the primary components, which are depicted in the image below.

C. Sequence Diagram



The sequence diagram of the hostel management system shows where admins will be able to log in to their account using their credentials. After logging in, the user can manage all the operations on the hostel: rooms, allotments, payments, and rent. All the pages, such as Allots, Payments, and Rent, are secure, and users can access them after logging in. The tile diagram below helps demonstrate how the tile login page works in a hostel management system. The various objects in the Payments, Hostel, Rooms, Allots, and Rent pages interact throughout the sequence, and the user will not be able to access this page without verifying their identity. This is the UML sequence diagram of the hostel management system, which shows the interaction between the objects of rooms, allotments, hostels, rent, and payments.

VI. FEASIBILITY STUDY

A. Technical Feasibility

Determining the technical feasibility of a proposed system depends on the system's technology, which encompasses the hardware and software used in the system. It also takes into account whether the technology used is up-to-date and whether the users will require any system modifications as advancements in technology occur. The host management system is technically feasible as it utilizes the Windows platform, an Apache XAMPP server, MySQL for the database, PHP as the programming language, and HTML or XML as the user interface.

B. Economic Feasibility

The most common approach to evaluating a new system's effectiveness is through cost-benefit analysis, which is an economic analysis. The internet provides easy access to PHP, HTML, XML, and MySQL databases.

VII. RESULT ANALYSIS

As a result, we have developed a way to make hostel booking more sophisticated utilizing technologies like HTML, CSS, JS, PHP, MySQL, and BOOTSTRAP in order to overcome the manual techniques. This project can be used as a hybrid; all the hostels can register themselves, and the students can book it according to their needs. Also, we have added grievance and filter functionality to our project. With a proper future scope, it will be good for use.

VIII. ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to everyone who has contributed to the success of this project. Our special thanks go to our guide, Prof. Rama Gaikwad, whose counselling was invaluable in ensuring that our project was focused and sound. She demonstrated a keen interest in checking even the minutest details of our work and provided valuable suggestions. Her guidance was not only technical but also moral. We are also grateful to Prof. Rama Gaikwad, our H.O.D., who provided us with direction and support throughout our project. We want to convey our thanks and appreciation to you. Our honourable principal, Dr. Sunil Thakre, provided us with the necessary resources and unwavering support.

CONCLUSION

Thus, we have developed PHP-MySQL source code, which would help in developing the online hostel management system. It will lessen the amount of work required to maintain the hostel for the owner and management. It will eliminate the need for traditional pen-and-paper methods. The developed system provides solutions to manual hostel management problems and also provides information such as hostel information, hostel room information, and hostel account information. It is an online software or website, so anyone can access it from anywhere without any complexity with the help of a working internet connection.

References

- [1] A study on shift towards digitization of Hostel Room Allotment for a university Author Tanmay Nandanwar and Priyanka Bahutule (2022).
- [2] Development of an Automated Hostel Facility Management System Journal of Science and Engineering Askar, Abhishek & Gaur, Saumya & Deolikar, Sejal & Ubale (2021).
- [3] Hostel Management System and Aggregation Online Hostel Management International Journal of Advanced Engineering & Science Research Prof Shyamsundar Magar, Ms. Sakshi Said and Mr. Rohit Jadhav (2021).
- [4] College Hostel Inventory Management System Sreerambabu, Dr Riyaz, Mr Rajkumar (2023).
- [5] Hostel Management System (HMS) Narkhede, Prof & Bamgude, Rutuja & Sonawane, Mayuri & Shevade, Mandar (2022).
- [6] Hostel Management System Based on Finger Print Authentication ISSN RAJKUMAR, G. & SUNDARI (2018).
- [7] Implementation Of Hostel Management with Automation Using Design Thinking the Research of PHP Development Framework Based on MVC Pattern Dinesh.B, Gogul Nithin.R, Pavatharani R,Sneha R,C.Senthilkumar (2022).
- [8] PHP Frameworks usability in Web Application Development Wei Cui,Lin Huang and LiJing Liang (2018).



International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: IJRPR 111455-3

Certificate of Acceptance & Publication

This certificate is awarded to "Sujal Suresh Thakre", and certifies the acceptance for publication of research paper entitled "Hostel Management System" in "International Journal of Research Publication and Reviews", Volume 4, Issue 12 .

Signed

Anurag Agarwal



Date

01-01-2024

Editor-in-Chief
International Journal of Research Publication and Reviews



International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582 7421

Sr. No: IJRPR 111455-4

Certificate of Acceptance & Publication

This certificate is awarded to "Sudaf Azam Sheikh", and certifies the acceptance for publication of research paper entitled "Hostel Management System" in "International Journal of Research Publication and Reviews", Volume 4, Issue 12 .

Signed

Anurag Agarwal



Date

01-01-2024

Editor-In-Chief
International Journal of Research Publication and Reviews



International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: URPR_111455-2

Certificate of Acceptance & Publication

This certificate is awarded to "Aditya Krishna Sapkal", and certifies the acceptance for publication of research paper entitled "Hostel Management System" in "International Journal of Research Publication and Reviews", Volume 4, Issue 12 .

Signed

Anand Agarwal



Date

01-01-2024

Editor-in-Chief
International Journal of Research Publication and Reviews



International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: URPR_111455-1

Certificate of Acceptance & Publication

This certificate is awarded to "Rachana Rajendra Patil", and certifies the acceptance for publication of research paper entitled "Hostel Management System" in "International Journal of Research Publication and Reviews", Volume 4, Issue 12 .

Signed

Anand Agarwal



Date

01-01-2024

Editor-in-Chief
International Journal of Research Publication and Reviews

HOSTEL MANAGEMENT SYSTEM

¹Prof.Pranjali More, ²Sujal Thakre, ³Sadaf Shaikh, ⁴Rachana Patil, ⁵Aditya Sapkal

¹Asst. prof., SPPU, Pune
^{2,3,4,5}Student, SPPU, Pune

Email: ¹pranjali.more@abmspcoerpune.org, ² sujalthakre123@gmail.com, ³ shaikhsadafazam@gmail.com,
⁴rachanapatil813@gmail.com, ⁵ sapkaladi1490@gmail.com

Abstract: The Hostel Management System with Attendance Tracking through Camera using Machine Learning is a comprehensive solution designed to streamline and enhance the efficiency of hostel management processes. The system employs advanced machine learning algorithms to automate attendance tracking through camera-based facial recognition, ensuring accurate and secure attendance records. The Hostel Management System with Attendance Tracking through Camera using Machine Learning is a sophisticated software solution designed to optimize the management of hostel facilities by integrating cutting-edge technologies. The primary focus of this system is to automate attendance tracking through facial recognition using machine learning algorithms.

Keywords: Hostel Management, Attendance Tracking, Facial Recognition.

I. INTRODUCTION

Using a manual system is not advised in the modern era of automated systems, whether they be hardware or software. Typically, hostels lacking a management system are closed once a year. Manual verification of the registration forms and other data recovery procedures is carried out, in many cases, by handwriting on paper. Therefore, it is possible to prevent a recurrence, using an automatic mechanism. A computerized system that will assist in reducing the amount of human input was designed in response to the shortcomings of the current methods. With this system in place, by improving the efficiency of the system, it is possible to improve on the weaknesses that are present under the current manual method. It is intended, with the hostel administration in mind, to allow student records of their rooms and belongings to be preserved. It's going to save them from the heavy manual work where it would be very difficult to find student records, college invoices, or any information on those who left the dorm in the first place.

II. IMPORTANCE OF TECHNOLOGY

A Hostel Management System (HMS) technology plays a crucial role in efficiently managing and organizing various aspects of hostel operations in educational institutions or other facilities. Here are some key points highlighting the importance of Hostel Management System technology:

Efficient Accommodation Management: MS automates the process of room allocation, ensuring optimal utilization of

hostel facilities. It helps in assigning rooms based on preferences, ensuring fairness and transparency.

Time and Cost Efficiency: Automation of administrative tasks like room allocation, fee collection, and attendance tracking reduces manual efforts and saves time. The system minimizes the need for paperwork, contributing to cost savings and environmental sustainability.

Improved Security: MS enhances security through features like biometric access control, CCTV monitoring, and visitor tracking. It helps in maintaining a secure environment by keeping a record of residents and visitors.

Attendance Tracking: The system allows for efficient tracking of hostel residents' attendance, ensuring compliance with rules and regulations. It helps identify and address any issues related to absenteeism.

Financial Management: Hostel Management System assists in the transparent and systematic collection of hostel fees. It provides accurate financial reports, helping management make informed decisions regarding budgeting and resource allocation.

Communication and Information Sharing: The system facilitates easy communication between hostel authorities, residents, and parents/guardians. Important information, updates, and announcements can be easily disseminated through the system.

Maintenance and Inventory Management: MS helps in scheduling and tracking maintenance activities for hostel

infrastructure. It assists in managing hostel inventories, ensuring that necessary supplies are available when needed.

Data Accuracy and Accessibility: The system maintains a centralized and accurate database of hostel-related information. Authorized personnel can access real-time data, improving decision-making processes.

Parental Involvement: Parents can have access to their child's hostel-related information, including attendance, performance, and communication from hostel authorities. This enhances transparency and fosters collaboration between parents and the hostel administration.

Regulatory Compliance: MS helps in ensuring compliance with regulatory requirements and standards governing hostel operations. It assists in generating reports for audits and inspections.

III. LITERATURE REVIEW

Begin with an overview of hostel management systems, highlighting their importance in efficiently managing hostel operations such as room allocation, fee management, student records, and staff management.

Explore the significance of implementing technology, like Java, Python, and AJAX, in developing a robust hostel management system.

Existing Hostel Management Systems:

Identify and review existing hostel management systems developed using Java, Python, or AJAX, or a combination of these technologies.

Evaluate their features, functionalities, user interfaces, scalability, and performance.

Critically analyze strengths, weaknesses, opportunities, and threats (SWOT analysis) of these systems.

Java in Hostel Management Systems:

Investigate how Java is utilized in developing hostel management systems.

Review literature discussing Java frameworks, libraries, and tools commonly used in HMS development.

Examine case studies or research papers demonstrating the application of Java in hostel management.

Python in Hostel Management Systems:

Explore the role of Python in hostel management systems.

Identify Python frameworks, such as Django or Flask, commonly used in HMS development.

Review studies or projects illustrating the integration of Python in hostel management solutions.

AJAX in Hostel Management Systems:

Discuss the significance of AJAX (Asynchronous JavaScript and XML) in enhancing the user experience and interactivity of hostel management systems.

Examine how AJAX is employed to create dynamic and responsive interfaces in HMS applications.

Review literature on AJAX best practices and its integration with Java and Python in web development.

Challenges and Solutions:

Identify common challenges faced in developing and deploying hostel management systems.

Evaluate proposed solutions and strategies to address these challenges.

Discuss research or projects focusing on overcoming technical, operational, or user-related hurdles in HMS implementation.

Future Trends and Directions:

Explore emerging trends and technologies shaping the future of hostel management systems.

Discuss potential advancements in Java, Python, AJAX, and related web development frameworks that could influence HMS development.

Consider the integration of machine learning, data analytics, or IoT (Internet of Things) in enhancing hostel management processes.

IV. RESEARCH METHODOLOGY

Define the Research Objectives: Clearly articulate the objectives of the research. Define what you aim to achieve with the implementation of a face recognition-based attendance system in the hostel.

Literature Review: Conduct a thorough literature review to understand the existing technologies, methodologies, and challenges related to hostel management systems and face recognition in attendance tracking. Identify gaps in current research that your project can address.

Identify Key Features and Requirements: List the essential features and requirements for the Hostel Management System with face recognition attendance tracking. This includes aspects such as accuracy, speed, adaptability to different environments, and integration capabilities.

Select ML Algorithms: Choose suitable ML algorithms for face recognition. Consider factors such as accuracy, training time, and resource requirements. Commonly used algorithms include Convolutional Neural Networks (CNNs) and Open Face.

Data Collection: Gather relevant data for training the ML model. This includes a dataset of facial images for recognition, as well as other data related to hostel management such as student details, room assignments, and historical attendance records.

Data Preprocessing: Preprocess the collected data to ensure it is clean, standardized, and suitable for training the ML model. This may involve image normalization, resizing, and feature extraction.

Model Training: Train the selected ML model using the prepared dataset. Fine-tune the model to achieve optimal performance in recognizing faces for attendance tracking.

System Development: Develop the Hostel Management System integrating the ML-based face recognition model. Ensure that the system meets the specified requirements and features identified in the earlier stages.

Testing and Validation: Conduct extensive testing to validate the accuracy and reliability of the face recognition system. Use a diverse set of test cases to simulate real-world scenarios and evaluate the system's performance.

User Feedback and Evaluation: Collect feedback from potential users, including hostel administrators and students. Evaluate the usability and user satisfaction with the implemented system.

Integration with Hostel Management System: Integrate the developed face recognition attendance tracking system with the broader Hostel Management System. Ensure smooth data flow and compatibility.

Performance Evaluation: Evaluate the overall performance of the system, considering factors such as speed, accuracy, and resource utilization. Compare the system's performance with traditional attendance tracking methods.

Security and Privacy Assessment: Assess the security measures in place to protect facial recognition data and ensure compliance with privacy regulations. Implement encryption and access controls as needed.

Documentation and Reporting: Document the entire research process, including the design, development, and testing phases. Prepare a comprehensive report outlining the methodology, results, and conclusions.

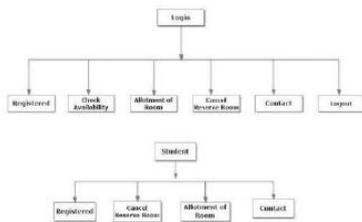


Figure 1: Flow Diagram of Proposed Work

V. FEASIBILITY STUDY

A. Technical Feasibility Determining the technical feasibility of a proposed system depends on the system's technology, which encompasses the hardware and software used in the system. It also takes into account whether the technology used is up-to-date and whether the users will require any system modifications as advancements in technology occur. The host management system is technically feasible as it utilizes the Windows platform, an Apache XAMPP server, MySQL for the database, PHP as the programming language, and HTML or XML as the user interface.

B. Economic Feasibility The most common approach to evaluating a new system's effectiveness is through cost-benefit analysis, which is an economic analysis. The

internet provides easy access to PHP, HTML, XML, and MySQL databases.

VI. REQUIREMENT ANALYSIS

A. Functional Requirements

1. The user shall generate a user profile containing the following information: user account number, full name, address, phone number, and room number.
2. Users will change the status of their dues in the database according to whether they are paid or not.
3. Users must allow the warden to add new users to the system's database.
4. The user must allow the guardian to cancel the registration from the system's database.

B. Non-Functional Requirements 1. The system shall be capable of operating reliably. It's always got to be raised up and running.

2. A high performance should be achieved by the system.

C. Hardware Requirements

A major task related to software development is the section on hardware configuration. The speed and efficiency of the whole system may be negatively affected by a lack of random-access memory. This process must be very strong enough to run all the operations. You should be able to store a program or file on your hard drive.

A. Size of cache: 512 KB. RAM: 512 MB (minimum)

B. Network Card: A 100 Mbps speed may be achieved with any card. Network connection: UTP, or coaxial cable connection Printer: An inkjet or laser-colour printer provides at least 1000 dpi. Hard disk: 80 GB Mouse: 104 keys (US Key, Serial, USB, or PS/2).

D. SOFTWARE CONFIGURATION

The compatibility section of the software is one of the most important parts of a system development, since it undergoes geometric evolution on the market. The chosen software should be compatible with the firm, as well as to one user, and it should be viable for the system. This document provides a comprehensive explanation of the software requirement specification. The study of requirement specification is focused especially on the functioning of the system. It enables the analyst or developer to comprehend the system, the task at hand, the desired performance level, and the associated interfaces that need to be set up

- Language to use: Python.
- Anaconda.
- IDE used: Vscode
- Operating system: Windows 11
-

VII. ADVANTAGES OF PROPOSED MODEL OVER EXISTING MODE

Enhanced Machine Learning Models: Continuously improve and update the facial recognition models to achieve higher accuracy and adaptability to diverse facial features and expressions. Explore advanced machine learning techniques, such as deep learning, for even more robust facial recognition capabilities.

Behavioural Analytics: Integrate behavioural analytics to identify patterns in residents' movements and interactions within the hostel. Analyse behaviour data for security purposes and to improve the overall hostel experience.

IoT Integration: Incorporate Internet of Things (IoT) devices to monitor and control aspects like room temperature, lighting, and energy consumption. Use IoT sensors for occupancy monitoring and optimizing resource usage in common areas.

Web Application Development: Develop web application for residents to access attendance records, receive notifications, and interact with hostel management services. Enable mobile-based attendance tracking using the camera on residents' smartphones.

Predictive Analytics: Implement predictive analytics to forecast future attendance trends based on historical data. Use predictive models to anticipate maintenance needs, optimize resource allocation, and enhance overall hostel planning.

VIII. RESULTS AND DISCUSSION

Automated Attendance Tracking: The system eliminates the need for manual attendance marking, reducing administrative workload and the likelihood of errors. Attendance records are accurately and automatically updated in real-time as residents enter or exit the hostel.

Improved Accuracy: Machine learning algorithms contribute to high accuracy in facial recognition, reducing the chances of fraudulent attendance. Continuous model training ensures adaptability to different facial features and environmental conditions.

Enhanced Security: Facial recognition adds an additional layer of security to hostel premises, preventing unauthorized access. Security alerts can be triggered in the case of suspicious activities or unrecognized individuals.

Efficient Administrative Processes: Hostel administrators can access a centralized dashboard to monitor attendance, allocate rooms, collect fees, and handle maintenance requests efficiently.

Automated notifications reduce the need for manual communication, saving time and resources.

Proactive Intervention: Real-time attendance updates enable administrators to identify absenteeism or tardiness promptly. Automated notifications to residents and administrators allow

for proactive intervention in case of irregular attendance patterns.

Data-driven Insights: The system generates reports and analytics on attendance trends, helping administrators make data-driven decisions. Historical attendance data can be analysed to identify patterns and optimize hostel management processes.

User Satisfaction: Residents benefit from a more streamlined and convenient attendance tracking process. Integration with a user-friendly interface enhances the overall hostel experience for both residents and administrators.

Cost and Resource Optimization: Automation reduces the reliance on manual labour for attendance tracking, allowing staff to focus on more strategic tasks. Optimal resource usage is facilitated through insights gained from attendance and occupancy data.

Scalability: The system is designed to scale with the growth of the hostel, accommodating an increasing number of residents and events. Integration capabilities with existing systems ensure scalability without disrupting ongoing operations.

Compliance with Regulations: Adherence to privacy and data protection regulations is ensured, contributing to legal compliance and building trust among residents.

Technological Innovation: The implementation showcases the hostel's commitment to embracing technological advancements for improved management processes. It positions the hostel as a modern and tech-savvy institution, attracting tech-conscious residents.

Positive Impact on Reputation: Successful implementation and positive outcomes contribute to the hostel's reputation for efficiency, security, and innovation. Word-of-mouth endorsements from satisfied residents can enhance the hostel's standing.



Figure 2: HOME PAGE

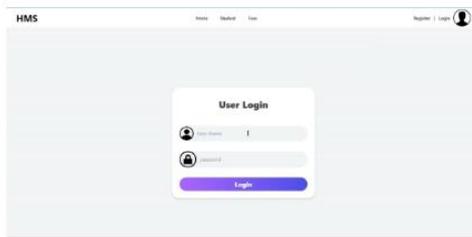


Figure 3: USER LOGIN



Figure 7: TOTAL ROOMS



Figure 4: IMAGE RECOGNITION

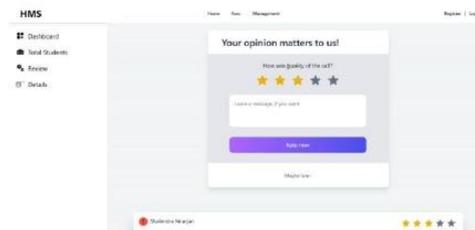


Figure 8: FEEDBACK

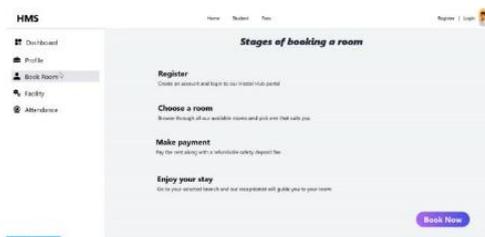


Figure 5: ROOM BOOKING



Figure 6: ATTENDANCE ANALYSIS

CONCLUSION

Thus, we have developed PHP-MySQL source code, which would help in developing the online hostel management system. It will lessen the amount of work required to maintain the hostel for the owner and management. It will eliminate the need for traditional pen-and-paper methods. The developed system provides solutions to manual hostel management problems and also provides information such as hostel information, hostel room information, and hostel account information. It is an online software or website, so anyone can access it from anywhere without any complexity with the help of a working internet connection.

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to everyone who has contributed to the success of this project. Our special thanks go to our guide, Prof. Rama Gaikwad, whose counselling was invaluable in ensuring that our project was focused and sound. She demonstrated a keen interest in checking even the minutest details of our work and provided valuable suggestions. Her guidance was not only technical but also moral. We are also grateful to Prof. Rama Gaikwad, our H.O.D., who provided us with direction and support throughout our project. We want to convey our thanks and appreciation to you. Our honourable principal, Dr. Sunil Thakre, provided us with the necessary resources and unwavering support.

REFERENCES

- [1] A study on shift towards digitization of Hostel Room Allotment for a university Author Tanmay Nandanwar and Priyanka Bahutule (2022).
- [2] Development of an Automated Hostel Facility Management System Journal of Science and Engineering Askar, Abhishek & Gaur, Saumya & Deolikar, Sejal & Ubale (2021).
- [3] Hostel Management System and Aggregation Online Hostel Management International Journal of Advanced Engineering & Science Research Prof. Shyamsundar Magar, Ms. Sakshi Said and Mr. Rohit Jadhav (2021).
- [4] College Hostel Inventory Management System Sreerambabu, Dr Riyaz, Mr Rajkumar (2023).
- [5] Hostel Management System (HMS) Narkhede, Prof & Bamgude, Rutuja & Sonawane, Mayuri & Shevade, Mandar (2022).
- [6] Hostel Management System Based on Finger Print Authentication ISSN RAJKUMAR, G. & SUNDARI (2018).
- [7] Implementation Of Hostel Management with
- [8] Automation Using Design Thinking the Research of Development Framework Based on MVC Pattern Dinesh.B, Gogul Nithin.R, Pavatharani.R,Sneha.R,C.Senthilkumar (2022).
- [9] Frameworks usability in Web Application Development Wei Cui,Lin Huang and LiJing Liang (2018).

□□□



National Conference
on

**Futuristic Trends in Engineering & Applied Science 2024
(NCFTEAS 2024)**

12th-13th April 2024

Organized by

VPM's Maharshi Parshuram College of Engineering, Velneswar

CERTIFICATE

OF PARTICIPATION

This is to certify that Rachana Rajendra Patil has presented a Paper entitled HOSTEL MANAGEMENT SYSTEM Co-authored with Sujal Thakre, Sadaf Shaikh, Aditya Sakpal, Rama Gaikwad in the National Conference on "Futuristic Trends in Engineering & Applied Science 2024 (NCFTEAS 2024) through offline/online mode held at VPM's MPCOE, Velneswar on 12th and 13th April 2024

Mr. Rahul Kundiya
Organizing Secretary

Mr. Avinash Pawar
Conference Chair

Dr. Narendra Soni
Conference Chair



Engineering Advances 2024: First International Conference_Abstract Accepted

2 messages

Chair Engineering Advances <contact@engineeringadvances.com>
To: 3251 Rachana Patil <rachanapatil813@gmail.com>

Sat, 20 Apr 2024 at 7:02 pm

Dear Author,
Congratulations !!

We are delighted to inform you that your abstract with the following details, submitted for the **Engineering Advances 2024: First International Conference** has been accepted for presentation at the conference.

Title

Hostel Management System

Authors

**Rachana Patil,
Sujal Thakre,
Aditya Sapkal,
Sadaf Shaikh
and Pranjali
More**

You will now need to,

1. Complete the Conference Registration process by paying the Early Bird Registration Fee (if paid on or before 25 Apr 2024) as below and also send the payment receipt to contact@engineeringadvances.com.

Oral Paper Presentation

Research Scholar: INR 1750

Academicians / Industry Professional: INR 2250

Poster Presentation

Students / Research Scholar: INR 900

Following are the account details for registration fee payment. You need to add the transaction charges as applicable.

Account Name: TECHNOLOGY RESEARCH AND INNOVATION CENTRE

Bank Account Number: 589401010050527

Name of the Bank: Union Bank of India

Account Type: Current

Branch Name: DSK Vishwa, Pune

Bank IFSC Code: UBIN0558940

You can also pay the registration fee by simply scanning the following QR code.



TECHNOLOGY RESEARCH AND INNOVA
triclindia@uboi

2. After paying the fees, please submit the [Conference Registration Form](#)

3. After registration, we will send you further paper / poster submission guidelines based on the information you filled in the Conference Registration Form.

You are also requested to visit the Engineering Advances 2024 website for further updates. In case of any further concerns, feel free to contact us.

With Warm Regards

Dr. Sandip A. Kale

Scientific Committee Chair

Engineering Advances 2024

www.engineeringadvances.com

3251 Rachana Patil <rachanapatil813@gmail.com>
To: patilrachana098@gamil.com

Mon, 29 Apr 2024 at 1:14 pm

[Quoted text hidden]

HOSTEL MANAGEMENT SYSTEM

¹Prof.Pranjali More, ²Sujal Thakre, ³Sadaf Shaikh, ⁴Rachana Patil, ⁵Aditya Sapkal

¹Asst. prof., SPPU, Pune
^{2,3,4,5}Student, SPPU, Pune

Email: ¹pranjali.more@abmspcorpune.org, ²sujalthakre123@gmail.com, ³shaikhsadafazam@gmail.com,
⁴rachanapatil813@gmail.com, ⁵sapkaladi1490@gmail.com

Abstract: The Hostel Management System with Attendance Tracking through Camera using Machine Learning is a comprehensive solution designed to streamline and enhance the efficiency of hostel management processes. The system employs advanced machine learning algorithms to automate attendance tracking through camera-based facial recognition, ensuring accurate and secure attendance records. The Hostel Management System with Attendance Tracking through Camera using Machine Learning is a sophisticated software solution designed to optimize the management of hostel facilities by integrating cutting-edge technologies. The primary focus of this system is to automate attendance tracking through facial recognition using machine learning algorithms.

Keywords: Hostel Management, Attendance Tracking, Facial Recognition.

I. INTRODUCTION

Using a manual system is not advised in the modern era of automated systems, whether they be hardware or software. Typically, hostels lacking a management system are closed once a year. Manual verification of the registration forms and other data recovery procedures is carried out, in many cases, by handwriting on paper. Therefore, it is possible to prevent a recurrence, using an automatic mechanism. A computerized system that will assist in reducing the amount of human input was designed in response to the shortcomings of the current methods. With this system in place, by improving the efficiency of the system, it is possible to improve on the weaknesses that are present under the current manual method. It is intended, with the hostel administration in mind, to allow student records of their rooms and belongings to be preserved. It's going to save them from the heavy manual work where it would be very difficult to find student records, college invoices, or any information on those who left the dorm in the first place.

II. IMPORTANCE OF TECHNOLOGY

A Hostel Management System (HMS) technology plays a crucial role in efficiently managing and organizing various aspects of hostel operations in educational institutions or other facilities. Here are some key points highlighting the importance of Hostel Management System technology:

Efficient Accommodation Management: MS automates the process of room allocation, ensuring optimal utilization of

hostel facilities. It helps in assigning rooms based on preferences, ensuring fairness and transparency.

Time and Cost Efficiency: Automation of administrative tasks like room allocation, fee collection, and attendance tracking reduces manual efforts and saves time. The system minimizes the need for paperwork, contributing to cost savings and environmental sustainability.

Improved Security: MS enhances security through features like biometric access control, CCTV monitoring, and visitor tracking. It helps in maintaining a secure environment by keeping a record of residents and visitors.

Attendance Tracking: The system allows for efficient tracking of hostel residents' attendance, ensuring compliance with rules and regulations. It helps identify and address any issues related to absenteeism.

Financial Management: Hostel Management System assists in the transparent and systematic collection of hostel fees. It provides accurate financial reports, helping management make informed decisions regarding budgeting and resource allocation.

Communication and Information Sharing: The system facilitates easy communication between hostel authorities, residents, and parents/guardians. Important information, updates, and announcements can be easily disseminated through the system.

Maintenance and Inventory Management: MS helps in scheduling and tracking maintenance activities for hostel

infrastructure. It assists in managing hostel inventories, ensuring that necessary supplies are available when needed.

Data Accuracy and Accessibility: The system maintains a centralized and accurate database of hostel-related information. Authorized personnel can access real-time data, improving decision-making processes.

Parental Involvement: Parents can have access to their child's hostel-related information, including attendance, performance, and communication from hostel authorities. This enhances transparency and fosters collaboration between parents and the hostel administration.

Regulatory Compliance: MS helps in ensuring compliance with regulatory requirements and standards governing hostel operations. It assists in generating reports for audits and inspections.

III. LITERATURE REVIEW

Begin with an overview of hostel management systems, highlighting their importance in efficiently managing hostel operations such as room allocation, fee management, student records, and staff management.

Explore the significance of implementing technology, like Java, Python, and AJAX, in developing a robust hostel management system.

Existing Hostel Management Systems:

Identify and review existing hostel management systems developed using Java, Python, or AJAX, or a combination of these technologies.

Evaluate their features, functionalities, user interfaces, scalability, and performance.

Critically analyze strengths, weaknesses, opportunities, and threats (SWOT analysis) of these systems.

Java in Hostel Management Systems:

Investigate how Java is utilized in developing hostel management systems.

Review literature discussing Java frameworks, libraries, and tools commonly used in HMS development.

Examine case studies or research papers demonstrating the application of Java in hostel management.

Python in Hostel Management Systems:

Explore the role of Python in hostel management systems.

Identify Python frameworks, such as Django or Flask, commonly used in HMS development.

Review studies or projects illustrating the integration of Python in hostel management solutions.

AJAX in Hostel Management Systems:

Discuss the significance of AJAX (Asynchronous JavaScript and XML) in enhancing the user experience and interactivity of hostel management systems.

Examine how AJAX is employed to create dynamic and responsive interfaces in HMS applications.

Review literature on AJAX best practices and its integration with Java and Python in web development.

Challenges and Solutions:

Identify common challenges faced in developing and deploying hostel management systems.

Evaluate proposed solutions and strategies to address these challenges.

Discuss research or projects focusing on overcoming technical, operational, or user-related hurdles in HMS implementation.

Future Trends and Directions:

Explore emerging trends and technologies shaping the future of hostel management systems.

Discuss potential advancements in Java, Python, AJAX, and related web development frameworks that could influence HMS development.

Consider the integration of machine learning, data analytics, or IoT (Internet of Things) in enhancing hostel management processes.

IV. RESEARCH METHODOLOGY

Define the Research Objectives: Clearly articulate the objectives of the research. Define what you aim to achieve with the implementation of a face recognition-based attendance system in the hostel.

Literature Review: Conduct a thorough literature review to understand the existing technologies, methodologies, and challenges related to hostel management systems and face recognition in attendance tracking. Identify gaps in current research that your project can address.

Identify Key Features and Requirements: List the essential features and requirements for the Hostel Management System with face recognition attendance tracking. This includes aspects such as accuracy, speed, adaptability to different environments, and integration capabilities.

Select ML Algorithms: Choose suitable ML algorithms for face recognition. Consider factors such as accuracy, training time, and resource requirements. Commonly used algorithms include Convolutional Neural Networks (CNNs) and Open Face.

Data Collection: Gather relevant data for training the ML model. This includes a dataset of facial images for recognition, as well as other data related to hostel management such as student details, room assignments, and historical attendance records.

Data Preprocessing: Preprocess the collected data to ensure it is clean, standardized, and suitable for training the ML model. This may involve image normalization, resizing, and feature extraction.

Model Training: Train the selected ML model using the prepared dataset. Fine-tune the model to achieve optimal performance in recognizing faces for attendance tracking.

System Development: Develop the Hostel Management System integrating the ML-based face recognition model. Ensure that the system meets the specified requirements and features identified in the earlier stages.

Testing and Validation: Conduct extensive testing to validate the accuracy and reliability of the face recognition system. Use a diverse set of test cases to simulate real-world scenarios and evaluate the system's performance.

User Feedback and Evaluation: Collect feedback from potential users, including hostel administrators and students. Evaluate the usability and user satisfaction with the implemented system.

Integration with Hostel Management System: Integrate the developed face recognition attendance tracking system with the broader Hostel Management System. Ensure smooth data flow and compatibility.

Performance Evaluation: Evaluate the overall performance of the system, considering factors such as speed, accuracy, and resource utilization. Compare the system's performance with traditional attendance tracking methods.

Security and Privacy Assessment: Assess the security measures in place to protect facial recognition data and ensure compliance with privacy regulations. Implement encryption and access controls as needed.

Documentation and Reporting: Document the entire research process, including the design, development, and testing phases. Prepare a comprehensive report outlining the methodology, results, and conclusions.

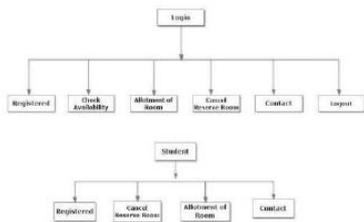


Figure 1: Flow Diagram of Proposed Work

V. FEASIBILITY STUDY

A. Technical Feasibility Determining the technical feasibility of a proposed system depends on the system's technology, which encompasses the hardware and software used in the system. It also takes into account whether the technology used is up-to-date and whether the users will require any system modifications as advancements in technology occur. The host management system is technically feasible as it utilizes the Windows platform, an Apache XAMPP server, MySQL for the database, PHP as the programming language, and HTML or XML as the user interface.

B. Economic Feasibility The most common approach to evaluating a new system's effectiveness is through cost-benefit analysis, which is an economic analysis. The

internet provides easy access to PHP, HTML, XML, and MySQL databases.

VI. REQUIREMENT ANALYSIS

A. Functional Requirements

1. The user shall generate a user profile containing the following information: user account number, full name, address, phone number, and room number.
2. Users will change the status of their dues in the database according to whether they are paid or not.
3. Users must allow the warden to add new users to the system's database.
4. The user must allow the guardian to cancel the registration from the system's database.

B. Non-Functional Requirements 1. The system shall be capable of operating reliably. It's always got to be raised up and running.

2. A high performance should be achieved by the system.

C. Hardware Requirements

A major task related to software development is the section on hardware configuration. The speed and efficiency of the whole system may be negatively affected by a lack of random-access memory. This process must be very strong enough to run all the operations. You should be able to store a program or file on your hard drive.

A. Size of cache: 512 KB. RAM: 512 MB (minimum)

B. Network Card: A 100 Mbps speed may be achieved with any card. Network connection: UTP, or coaxial cable connection Printer: An inkjet or laser-colour printer provides at least 1000 dpi. Hard disk: 80 GB Mouse: 104 keys (US Key, Serial, USB, or PS/2).

D. SOFTWARE CONFIGURATION

The compatibility section of the software is one of the most important parts of a system development, since it undergoes geometric evolution on the market. The chosen software should be compatible with the firm, as well as to one user, and it should be viable for the system. This document provides a comprehensive explanation of the software requirement specification. The study of requirement specification is focused especially on the functioning of the system. It enables the analyst or developer to comprehend the system, the task at hand, the desired performance level, and the associated interfaces that need to be set up

- Language to use: Python.
- Anaconda.
- IDE used: Vscode
- Operating system: Windows 11
-

VII. ADVANTAGES OF PROPOSED MODEL OVER EXISTING MODE

Enhanced Machine Learning Models: Continuously improve and update the facial recognition models to achieve higher accuracy and adaptability to diverse facial features and expressions. Explore advanced machine learning techniques, such as deep learning, for even more robust facial recognition capabilities.

Behavioural Analytics: Integrate behavioural analytics to identify patterns in residents' movements and interactions within the hostel. Analyse behaviour data for security purposes and to improve the overall hostel experience.

IoT Integration: Incorporate Internet of Things (IoT) devices to monitor and control aspects like room temperature, lighting, and energy consumption.

Use IoT sensors for occupancy monitoring and optimizing resource usage in common areas.

Web Application Development: Develop web application for residents to access attendance records, receive notifications, and interact with hostel management services. Enable mobile-based attendance tracking using the camera on residents' smartphones.

Predictive Analytics: Implement predictive analytics to forecast future attendance trends based on historical data. Use predictive models to anticipate maintenance needs, optimize resource allocation, and enhance overall hostel planning.

VIII. RESULTS AND DISCUSSION

Automated Attendance Tracking: The system eliminates the need for manual attendance marking, reducing administrative workload and the likelihood of errors. Attendance records are accurately and automatically updated in real-time as residents enter or exit the hostel.

Improved Accuracy: Machine learning algorithms contribute to high accuracy in facial recognition, reducing the chances of fraudulent attendance.

Continuous model training ensures adaptability to different facial features and environmental conditions.

Enhanced Security: Facial recognition adds an additional layer of security to hostel premises, preventing unauthorized access. Security alerts can be triggered in the case of suspicious activities or unrecognized individuals.

Efficient Administrative Processes: Hostel administrators can access a centralized dashboard to monitor attendance, allocate rooms, collect fees, and handle maintenance requests efficiently.

Automated notifications reduce the need for manual communication, saving time and resources.

Proactive Intervention: Real-time attendance updates enable administrators to identify absenteeism or tardiness promptly. Automated notifications to residents and administrators allow

for proactive intervention in case of irregular attendance patterns.

Data-driven Insights: The system generates reports and analytics on attendance trends, helping administrators make data-driven decisions. Historical attendance data can be analysed to identify patterns and optimize hostel management processes.

User Satisfaction: Residents benefit from a more streamlined and convenient attendance tracking process. Integration with a user-friendly interface enhances the overall hostel experience for both residents and administrators.

Cost and Resource Optimization: Automation reduces the reliance on manual labour for attendance tracking, allowing staff to focus on more strategic tasks. Optimal resource usage is facilitated through insights gained from attendance and occupancy data.

Scalability: The system is designed to scale with the growth of the hostel, accommodating an increasing number of residents and events. Integration capabilities with existing systems ensure scalability without disrupting ongoing operations.

Compliance with Regulations: Adherence to privacy and data protection regulations is ensured, contributing to legal compliance and building trust among residents.

Technological Innovation: The implementation showcases the hostel's commitment to embracing technological advancements for improved management processes. It positions the hostel as a modern and tech-savvy institution, attracting tech-conscious residents.

Positive Impact on Reputation: Successful implementation and positive outcomes contribute to the hostel's reputation for efficiency, security, and innovation. Word-of-mouth endorsements from satisfied residents can enhance the hostel's standing.



Figure 2: HOME PAGE

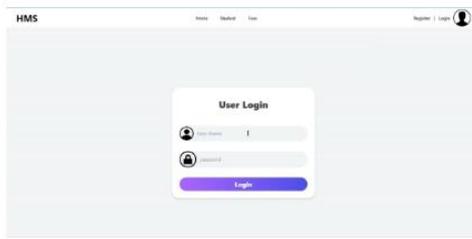


Figure 3: USER LOGIN

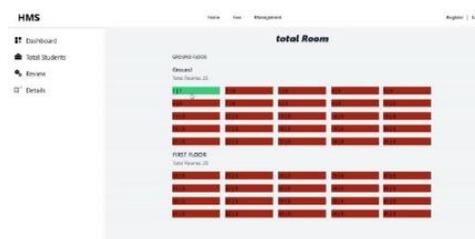


Figure 7: TOTAL ROOMS



Figure 4: IMAGE RECOGNITION

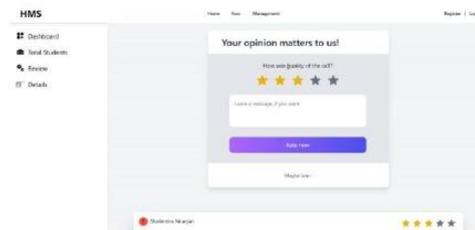


Figure 8: FEEDBACK

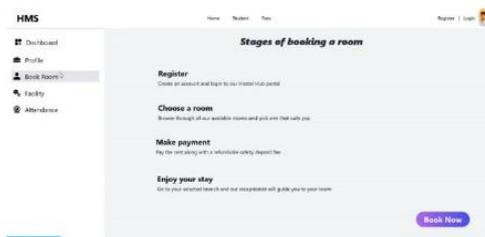


Figure 5: ROOM BOOKING

CONCLUSION

Thus, we have developed PHP-MySQL source code, which would help in developing the online hostel management system. It will lessen the amount of work required to maintain the hostel for the owner and management. It will eliminate the need for traditional pen-and-paper methods. The developed system provides solutions to manual hostel management problems and also provides information such as hostel information, hostel room information, and hostel account information. It is an online software or website, so anyone can access it from anywhere without any complexity with the help of a working internet connection.



Figure 6: ATTENDANCE ANALYSIS

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to everyone who has contributed to the success of this project. Our special thanks go to our guide, Prof. Rama Gaikwad, whose counselling was invaluable in ensuring that our project was focused and sound. She demonstrated a keen interest in checking even the minutest details of our work and provided valuable suggestions. Her guidance was not only technical but also moral. We are also grateful to Prof. Rama Gaikwad, our H.O.D., who provided us with direction and support throughout our project. We want to convey our thanks and appreciation to you. Our honourable principal, Dr. Sunil Thakre, provided us with the necessary resources and unwavering support.

REFERENCES

- [1] A study on shift towards digitization of Hostel Room Allotment for a university Author Tanmay Nandanwar and Priyanka Bahutule (2022).
- [2] Development of an Automated Hostel Facility Management System Journal of Science and Engineering Askar, Abhishek & Gaur, Saumya & Deolikar, Sejal & Ubale (2021).
- [3] Hostel Management System and Aggregation Online Hostel Management International Journal of Advanced Engineering & Science Research Prof. Shyamsundar Magar, Ms. Sakshi Said and Mr. Rohit Jadhav (2021).
- [4] College Hostel Inventory Management System Sreerambabu, Dr Riyaz, Mr Rajkumar (2023).
- [5] Hostel Management System (HMS) Narkhede, Prof & Bamgude, Rutuja & Sonawane, Mayuri & Shevade, Mandar (2022).
- [6] Hostel Management System Based on Finger Print Authentication ISSN RAJKUMAR, G. & SUNDARI (2018).
- [7] Implementation Of Hostel Management with
- [8] Automation Using Design Thinking the Research of Development Framework Based on MVC Pattern Dinesh,B, Gogul Nithin,R, Pavatharani,R,Sneha,R,C.Senthilkumar (2022).
- [9] Frameworks usability in Web Application Development Wei Cui,Lin Huang and LiJing Liang (2018).

□□□



CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Aditya Sapkal

for presenting the paper entitled
HOSTEL MANAGEMENT SYSTEM

at the International Conference on Advanced Technologies and Intelligent Automation (ICATIA-24), 18th-19th May 2024 hosted by Protean Tech Services, Pune, India



55613213377796

CERTIFICATE ID

19.05.2024



Dr. Mathew Vkk
PROTEAN TECH SERVICES



CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Rachana Patil

for presenting the paper entitled
Hostel management system

at the International Conference on Advanced Technologies and Intelligent Automation (ICATIA-24), 18th-19th May 2024 hosted by Protean Tech Services, Pune, India



80125524682143

CERTIFICATE ID

19.05.2024



Dr. Mathew Vkk
PROTEAN TECH SERVICES



CERTIFICATE

OF APPRECIATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Sadaf Azam Shaikh

for presenting the paper entitled
Hostel Management System

at the International Conference on Advanced Technologies and Intelligent Automation (ICATIA-24), 18th-19th May 2024 hosted by Protean Tech Services, Pune, India



91905487443300

CERTIFICATE ID
20.05.2024



Dr. Mathew Vkk

PROTEAN TECH SERVICES



CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Sujal Suresh Thakre

for presenting the paper entitled
Hostel Management System

at the International Conference on Advanced Technologies and Intelligent Automation (ICATIA-24), 18th-19th May 2024 hosted by Protean Tech Services, Pune, India



73210588851940

CERTIFICATE ID
19.05.2024



Dr. Mathew Vkk

PROTEAN TECH SERVICES

Review:



Reg: Acceptance of Abstract ICATIA24 - CST 1111

1 message

ICATIA-2024 <icatia2024@easychair.org>
To: Rachana Patil <rachanapatil813@gmail.com>

Sun, 25 Feb 2024 at 11:16 pm

Dear Authors,

Congratulations !!! We are pleased to inform you that your abstract has been accepted for presentation at the International Conference on Advanced Technologies and Intelligent Automation (ICATIA24), scheduled to be held online on 18th-19th May 2024.

Please note your Paper ID for Future Reference: ICATIA24 - CST 1111.

We appreciate your interest in contributing to ICATIA 2024 and look forward to your participation in the conference. To confirm your participation, please follow these steps:

Registration: Register for the conference by paying the required fees. You can find the registration fees and payment instructions on the conference website: [<https://icatia.in/registration>].

Form Submission: Fill out the online registration form available on the conference website. [<https://forms.gle/5ab16M58TqpisxCr5>]

Full Paper Submission: Submit your full-length paper by April 15, 2024. The paper should be submitted electronically through the conference website. Please ensure that your paper adheres to the conference formatting guidelines, which can be found on the website. [<https://icatia.in/author-guidelines>].

Full length paper submission : <https://forms.gle/2fK1nFHZvJA2P44R7>

Important Dates:

Registration Deadline: 1st March 2024

Full Paper Submission Deadline: April 15, 2024

Review Process: Your paper will undergo a rigorous review process by a panel of experts. You will be notified of the review decision within a week after your submission.

Revised Paper Submission Deadline: You will be required to submit a revised version by May 15, 2024.

As Mentioned earlier, UG Students are eligible for the Free Internship. For more details please visit the Website

We encourage you to submit your full paper as early as possible to allow for a thorough review process and provide you with ample time to revise your paper if necessary.

For any questions or clarifications, please do not hesitate to contact us at icatia@proteantechservices.in.

We look forward to welcoming you to ICATIA 2024!

All Authors Please ensure you Join the Group for further Updates. Join WhatsApp group: <https://chat.whatsapp.com/EzZHtnFBMyBE4K0pF1v0st>

ANNEXTURE D
PLAGIARISM REPORT

Plagiarism Scan Report



Characters:1497	Words:187
Sentences:11	Speak Time: 2 Min

Excluded URL None

Content Checked for Plagiarism

1.1 PROJECT TITLE Hostel management system 1.2 PROJECT OPTION Basic Attendance Tracking System: Develop a simple hostel management system focused primarily on attendance tracking using basic technologies such as RFID or barcode scanners. Ideal for small-scale hostel facilities or as a starting point for learning about hostel management systems. Camera-Based Attendance System: Implement a more advanced hostel management system utilizing camera attendance technology for automated attendance tracking. Incorporate facial recognition algorithms for accurate identification and recording of student attendance. Suitable for mid to large-scale hostel facilities looking to enhance security and efficiency. Comprehensive Hostel Management System: Develop a fully-featured hostel management system covering various aspects such as room allocation, inventory management, billing, and reporting. Include modules for student registration, room assignment, maintenance tracking, and staff management. Targeted towards larger hostel facilities or institutions with complex management needs. Mobile Application Integration: Create a mobile application interface for the hostel management system, allowing students to check-in/out, view room assignments, and receive notifications. Enhance accessibility and convenience for students and staff, enabling them to access hostel-related information from their smartphones. Suitable for modernizing hostel management processes and improving user experience.

Sources



[Home](#) [Blog](#) [Testimonials](#) [About Us](#) [Privacy Policy](#)

Copyright © 2024 [Plagiarism Detector](#). All right reserved

Plagiarism Scan Report



Characters:2389	Words:352
Sentences:16	Speak Time: 3 Min

Excluded URL None

Content Checked for Plagiarism

1.1 INTERNAL GUIDE Prof. Pranjali More 1.2 TECHNICAL KEYWORDS Allocation: The term "allocation" refers to the process of distributing resources, tasks, or responsibilities among various parties or uses. It involves assigning specific amounts or portions of something, such as money, time, materials, or labor, to different people, projects, or categories based on certain criteria or priorities. Database: A database is an organized collection of structured information or data, typically stored electronically in a computer system. File Management: File management refers to the process of organizing, storing, naming, and handling files on a computer or a digital storage system. Effective file management ensures that files are easily accessible, secure, and properly maintained. 1.3 PROBLELM STATEMENT Design and develop a Hostel Management System to address the challenges faced by a large hostel facility in efficiently managing its operations. The hostel currently faces several manual and time-consuming processes, leading to inefficiencies, data inaccuracies, and reduced guest satisfaction. The objective is to create a comprehensive software solution that automates key tasks and enhances overall hostel management. 1.4 ABSTRACT This project is based on the Design and Implementation of an Online Hotel Management System for use at the Mountain Top University Hostel. It evolved from the manual file record system used to store hostel details, room details, student records, hostel allocation and other data in the hostel. With the help of the internet, computers and other mobile devices are used to manage hostel records and help the administrator effectively access these records. The goal of this project is to develop a dependable and effective hostel administration system that can easily handle all manual labor. To accomplish its goals, an incremental and iterative design process was used, along with the creation of a database. This project is carried out using PHP, MySQL with other frontend and backend technologies using Visual Studio Community 2017 as the Integrated Development Environment. Record officers and hostel managers may better monitor and oversee their separate activities with the use of the developed Hostel Management System. Additionally, it is a quicker and more effective approach to maintain and keep an eye on hostel data

Sources

6% Plagiarized

Plagiarism Scan Report



Characters:1980	Words:260
Sentences:15	Speak Time: 3 Min

Excluded URL None

Content Checked for Plagiarism

2. Algorithm Development · Graph Theory: Graph Representation: Rooms, students, and pathways can be represented as graphs for optimized allocation and navigation. Search Algorithms: Breadth-First Search (BFS) and Depth-First Search (DFS) for finding available rooms or resources. · Sorting and Searching Algorithms: Binary Search: Efficiently finding a student record or room. Sorting Algorithms: QuickSort, MergeSort for ordering data (e.g., by fee due dates or room numbers). 3. Data Analysis · Descriptive Statistics: Mean, Median, Mode: For analyzing student data like age, room occupancy rates, or fee payments. Standard Deviation and Variance: Understanding the dispersion in data such as fee payment patterns or room occupancy. · Inferential Statistics: Hypothesis Testing: Determining if changes (like a new fee structure) have statistically significant effects on payments. Regression Analysis: Predicting trends like future occupancy rates based on historical data. 4. Financial Calculations · Basic Arithmetic and Algebra: Addition, Subtraction, Multiplication, Division: For fee calculations, total payments, and outstanding balances. Percentage Calculations: Discounts, late fee percentages, and interest rates. 5. Optimization Techniques · Linear Programming: Objective Function: Minimize costs or maximize room occupancy. Constraints: Budget limits, room capacities, maintenance schedules. 6. Data Visualization · Coordinate Geometry: Graphs and Charts: Plotting occupancy trends, fee collection over time, and maintenance requests. · Matrix Algebra: Transformation and Rotation: For graphical representations and room layout planning. Practical Application in Project Database Schema Design: Use relational algebra concepts to design tables and relationships. Implement efficient query operations using SQL. Room Allocation Algorithm: Use graph theory to model rooms and optimize allocation. Implement sorting and searching algorithms for quick data retrieval.

Sources



[Home](#) [Blog](#) [Testimonials](#) [About Us](#) [Privacy Policy](#)

Copyright © 2024 [Plagiarism Detector](#). All right reserved

Plagiarism Scan Report



Characters:1901	Words:238
Sentences:9	Speak Time: 2 Min

Excluded URL None

Content Checked for Plagiarism

1.1 GOALS AND OBJECTIVES · Enhance Transparency: Implement blockchain technology to provide a transparent and auditable ledger of crowdfunding transactions, ensuring trust and accountability among participants. · Reduce Costs: Minimize fees associated with traditional crowdfunding platforms by eliminating intermediaries and automating processes through smart contracts, thereby increasing the funds available for projects. · Expand Access: Facilitate global participation in crowdfunding campaigns by removing geographical barriers and enabling contributions in various cryptocurrencies, fostering inclusivity and diversity among backers. · Mitigate Fraud: Utilize blockchain's immutability and security features to reduce the risk of fraudulent activities such as misappropriation of funds or manipulation of campaign data, safeguarding the interests of both backers and project creators. · Improve Efficiency: Streamline crowdfunding processes through smart contracts, automating tasks such as fund disbursement, milestone tracking, and voting mechanisms, enhancing efficiency and reducing administrative overhead. · Ensure Compliance: Navigate regulatory complexities and ensure compliance with relevant laws and regulations governing crowdfunding activities in different jurisdictions, fostering a legal framework conducive to decentralized fundraising. · Empower Innovation: Empower entrepreneurs, startups, and creators to access capital and bring their ideas to life by providing a decentralized and inclusive platform for fundraising, fostering innovation and economic growth. 1.2 RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT 1. Database Management Relational Algebra: Set Theory: Used for operations like UNION, INTERSECT, and DIFFERENCE on tables. Cartesian Product: Combining tuples from two relations. Selection and Projection: Filtering rows and columns in a database.

Sources



[Home](#) [Blog](#) [Testimonials](#) [About Us](#) [Privacy Policy](#)

Copyright © 2024 [Plagiarism Detector](#). All right reserved

**ANNEXTURE E
INFORMATION OF
PROJECT GROUP
MEMBERS**

Name: Sujal Suresh Thakre

Date of Birth: 08/09/2002

Gender: Male

Permanent Address: Akola

Email: sujalthakre123@gmail.com

Mobile/Contact No. :7249698981

Placement Details: Not Yet

Paper Publish: International Conference on Advanced Technologies and Intelligent Automation



Name: Sadaf Azam Shaikh

Date of Birth: 28/08/2002

Gender: Female

Permanent Address: Pune

Email: shaikhsadafazam@gmail.com

Mobile/Contact No. :8668444758

Placement Details: Not Yet

Paper Publish: International Conference on Advanced Technologies and Intelligent Automation



Name: Aditya Krishna Sapkal

Date of Birth: 06/10/2001

Gender: Male

Permanent Address: Jalgaon Sapkal

Email: sapkaladi1490@gmail.com

Mobile/Contact No. :9921910754

Placement Details: Not Yet

Paper Publish: International Conference on Advanced
Technologies and Intelligent Automation



Name: Rachana Rajendra Patil

Date of Birth: 16/01/2002

Gender: Female

Permanent Address: Shevgaon

Email: rachanapatil813@gmail.com

Mobile/Contact No. :8847781497

Placement Details: Not Yet

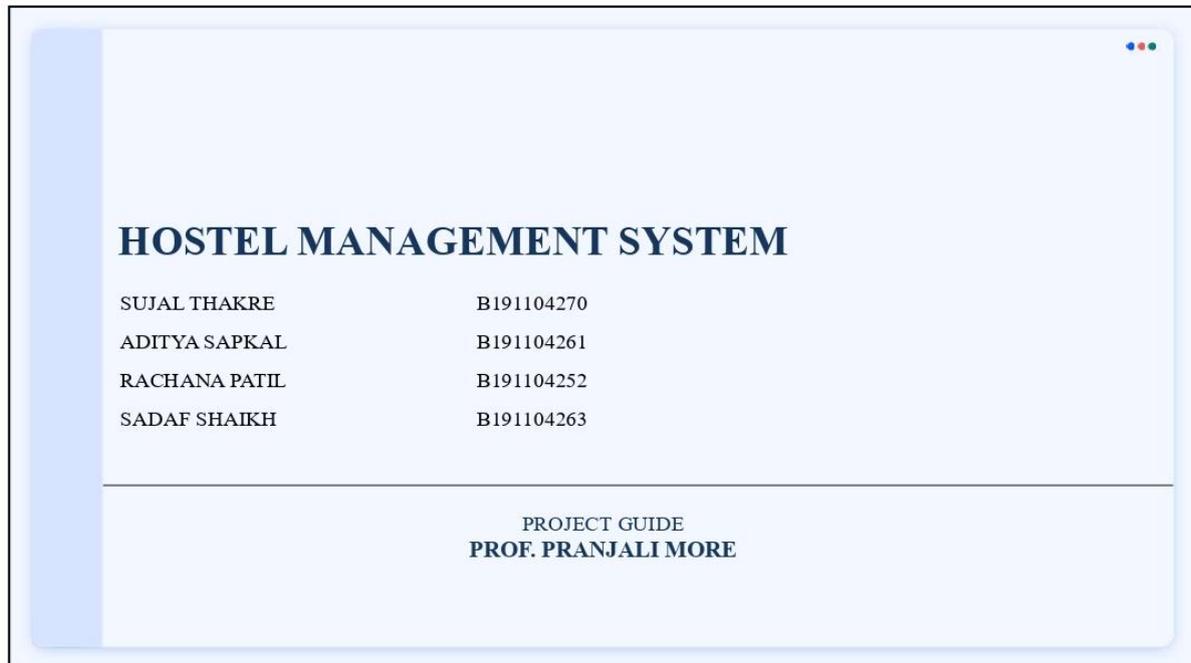
Paper Publish: International Conference on Advanced
Technologies and Intelligent Automation



ANNEXTURE F

FINAL PROJECT

PRESENTATION HANDOUTS

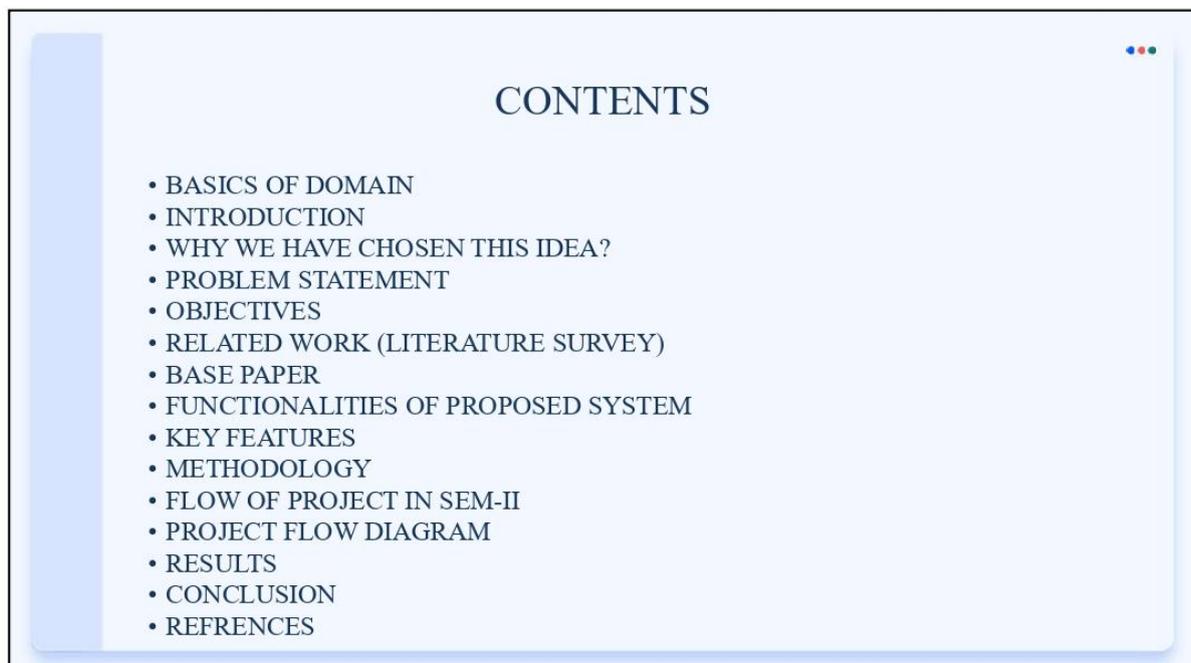


HOSTEL MANAGEMENT SYSTEM

SUJAL THAKRE	B191104270
ADITYA SAPKAL	B191104261
RACHANA PATIL	B191104252
SADAF SHAIKH	B191104263

PROJECT GUIDE
PROF. PRANJALI MORE

1



CONTENTS

- BASICS OF DOMAIN
- INTRODUCTION
- WHY WE HAVE CHOSEN THIS IDEA?
- PROBLEM STATEMENT
- OBJECTIVES
- RELATED WORK (LITERATURE SURVEY)
- BASE PAPER
- FUNCTIONALITIES OF PROPOSED SYSTEM
- KEY FEATURES
- METHODOLOGY
- FLOW OF PROJECT IN SEM-II
- PROJECT FLOW DIAGRAM
- RESULTS
- CONCLUSION
- REFERENCES

2

1

BASICS OF DOMAIN

Data Management:

Residents: Information about students or occupants, including details like names, contact information, and room assignments.

Rooms & Beds: Management of rooms, their types (single, double), facilities, and available beds.

Processes:

Admissions & Allotment: Streamlining the application process, managing eligibility criteria, and assigning rooms to residents.

Attendance: Monitoring resident movement and tracking attendance (if applicable).

Reports & Analytics: Generating reports on occupancy rates, fees collected, and other relevant data.

3

INTRODUCTION

HOSTEL MANAGEMENT SYSTEM is a Machine Learning based software developed for managing various activities in the hostel.

The hostel's day-to-day operations are increasing , to cater to the growing number of students enrolled in this institution. Consequently, those overseeing the hostels are under considerable pressure, typically without the aid of software. This specific project addresses the challenges of hostel management, aiming to alleviate the issues associated with manual processes.

It is intended to make the management processes more efficient , organized , and user friendly to both hostel administrators and students.

4

2

WHY WE HAVE CHOSEN THIS IDEA?

- 1. Application
- 2. Automation and Efficiency
- 3. Skill Development
- 4. Scalability and Flexibility
- 5. Problem-Solving
- 6. User Impact
- 7. Data Management and Reporting
- 8. Integration with Other Systems
- 9. User-Friendly Experience

5

PROBLEM STATEMENT

Design and develop a Hostel Management System to address the challenges faced by a large hostel facility in efficiently managing its operations. The hostel currently faces several manual and time-consuming processes, leading to inefficiencies, data inaccuracies, and reduced student satisfaction. The objective is to create a comprehensive software solution that automates key tasks and enhances overall hostel management.

6

3

OBJECTIVES

Efficient Accommodation:
Ensure efficient and fair allocation of hostel rooms to college students based on their preferences and availability.

Streamlined Processes:
Automate various tasks, such as booking, billing, and maintenance, to streamline hostel management processes.

Enhanced Student Experience:
Provide a seamless attendance process, convenient online booking, and prompt grievance redressal to enhance the overall student experience.

Accurate Financial Management:
Ensure accurate billing and payment tracking to manage hostel finances effectively.

Data Management and Reporting:
Maintain comprehensive student records and generate reports to facilitate data-driven decision-making and planning.

Improved Security:
Implement access control measures to ensure the safety and security of students residing in the hostel.

7

RELATED WORK (LITERATURE SURVEY)

Sr. no	Author	year	Advantages	Disadvantages
1	Harina P Kavya K K Sharmikha Sree R Meera S	2022	Better resource management: Hostel management systems aid in the effective management of resources such as food, utilities, and other amenities, leading to cost savings and improved service delivery.	Data Security Concerns: Storing sensitive information in the system can pose data security risks, necessitating robust security measures to prevent breaches and leaks.
2	Prof. Shyamsundar Magar Mr. Rohit Jadhav Ms. Sakshi Said Mr. Shashikant Jadhav	2021	Efficient room allocation: Hostel management systems enable optimized room allocation, ensuring that available space is utilized effectively	Initial Cost: Implementing a hostel management system can involve significant upfront costs for software, hardware, and training
3	Mr. Kathiresh Ms. Nandika Mr. Naveenkumar Mr. Manikandan	2023	Streamlined communication: These systems facilitate seamless communication between hostel staff, students, and administrators, enhancing overall coordination and reducing miscommunication	Technical Issues: System glitches, software bugs, or hardware malfunctions can disrupt operations and cause inconvenience to both staff and residents

8

4	Abhishek Askar, Saumya Gaur, Sejal Deoliker And Mrs. Archana Ubale	2024	Enhanced parent-student communication: Through features such as online portals and regular updates, hostel management systems facilitate better communication between parents and hostel authorities, providing peace of mind for both parties	Dependency on Technology: Reliance on technology makes the system vulnerable to potential outages or cybersecurity threats, leading to downtime and data breaches.
5	Dr. J. Sree Rambabu Mr. M. Mohammed Riyaz Mr. D. Rajkumar Mr. Tamilarasan	2023	Real-time monitoring: With the capability for real-time monitoring of hostel activities and operations, these systems enable prompt intervention in case of any issues or emergencies	Training Requirements: Staff members may require extensive training to effectively use the management system, which can be time-consuming and costly
6	Kartik Chaudhri, Riddhi Kevat	2021	Enhanced parent-student communication: Through features such as online portals and regular updates, hostel management systems facilitate better communication between parents and hostel authorities, providing peace of	Lack of Customization: Some system may lack the flexibility to accommodate specific hostel requirements, leading to limitations in customization.

9

BASE PAPER

The paper titled "Design and Implementation of Hostel Management System using Java and MySQL" , which appeared in LC INTERNATIONAL JOURNAL OF STEM , December 2020 is selected as Base paper for this project.

It proposes the development of a simple Hostel Management System in Java , backed by MySQL database. The system has a simple , easy to understand GUI , with modules for administrator , warden and student.

Limitations : This implementation does not provide a functionality to capture, monitor or store attendance of students. The support from database is quite limited and might not withstand the increasing number of records ,along with the visual data. Moreover , the look and feel of the system can be improved drastically , using more advanced front-end technologies.

10

5

FUNCTIONALITIES OF PROPOSED SYSTEM

Enrollment and attendance Process:
Streamline the registration and attendance process for students using face recognition, including verification of student details.

Student Accommodation Management:
Efficiently manage room allocations, availability, and occupancy status for college students staying in the hostel.

Room Reservation and Booking:
Allow students to book hostel rooms online, check availability, and receive confirmation of their reservations.

13

FUNCTIONALITIES OF PROPOSED SYSTEM

Student Profiles and Records:
Maintain comprehensive profiles of students, including personal information, contact details, and hostel stay history.

14

FUNCTIONALITIES OF PROPOSED SYSTEM

Student Profiles and Records:
Maintain comprehensive profiles of students, including personal information, contact details, and hostel stay history.

Efficient Billing System:
Automate the billing process, calculate hostel fees and track payment status for each student.

15

FUNCTIONALITIES OF PROPOSED SYSTEM

Student Profiles and Records:
Maintain comprehensive profiles of students, including personal information, contact details, and hostel stay history.

Efficient Billing System:
Automate the billing process, calculate hostel fees and track payment status for each student.

Complaints and Feedback System:
Provide a platform for students to submit complaints and feedback regarding hostel facilities and services.

16

KEY FEATURES

Daily attendance through Face detection :
Automates the attendance process using face recognition technology, eliminating the need for manual attendance marking.

Enhanced Security: Integrate face recognition as a secure authentication method for access control within the hostel premises, ensuring only authorized individuals are granted entry.

Efficient Room Allocation:
Develop algorithms to optimize room allocation based on resident preferences and availability, ensuring efficient utilization of hostel resources.

User-Friendly Interface:
An intuitive and user-friendly interface for both administrators and residents, making it easy to navigate and perform necessary tasks within the system.

Resource Optimization:
Optimize room occupancy and inventory management to make the best use of hostel resources.

21

KEY FEATURES

Daily attendance through Face detection :
Automates the attendance process using face recognition technology, eliminating the need for manual attendance marking.

Enhanced Security: Integrate face recognition as a secure authentication method for access control within the hostel premises, ensuring only authorized individuals are granted entry.

Efficient Room Allocation:
Develop algorithms to optimize room allocation based on resident preferences and availability, ensuring efficient utilization of hostel resources.

User-Friendly Interface:
An intuitive and user-friendly interface for both administrators and residents, making it easy to navigate and perform necessary tasks within the system.

Resource Optimization: Optimize room occupancy and inventory management to make the best use of hostel resources.

Scalability and Growth:
Develop a scalable system that can accommodate the growing number of students and future expansion of hostel facilities.

22

11

METHODOLOGY

- Planning and Requirement Gathering
 - Identify the key stakeholders (hostel administrators, students, staff, etc.)
 - Conduct meetings, interviews, and workshops to gather requirements
 - Create user stories that capture the requirements from the stakeholders' perspectives
 - Prioritize the user stories based on business value and importance
- Sprint Planning
 - Break down the project into smaller, manageable iterations called sprints (typically 2-4 weeks long)
 - Select a subset of high-priority user stories for each sprint
 - Define the sprint goal and create a sprint backlog with tasks and estimates

d

23

METHODOLOGY

- Sprint Execution
 - Hold daily stand-up meetings to discuss progress, challenges, and plan for the day
 - Develop and implement the features based on the user stories in the sprint backlog
 - Conduct continuous integration and testing throughout the sprint
 - Collaborate and communicate regularly with the team and stakeholders
- Sprint Review and Retrospective
 - At the end of each sprint, conduct a sprint review meeting
 - Demonstrate the completed work to the stakeholders and gather feedback
 - Conduct a sprint retrospective to identify areas for improvement and lessons learned
 - Update the product backlog with new or revised user stories based on feedback

24

12

METHODOLOGY

Throughout the Agile process, the following practices are commonly used:

Continuous Integration and Testing: Frequently integrate code changes and automatically run tests to catch issues early.

Pair Programming: Two developers collaborate on writing code to improve code quality and knowledge sharing.

Refactoring: Restructure the codebase to improve its internal structure without changing its external behavior.

Continuous Deployment: Automate the deployment process to quickly release new features and updates.

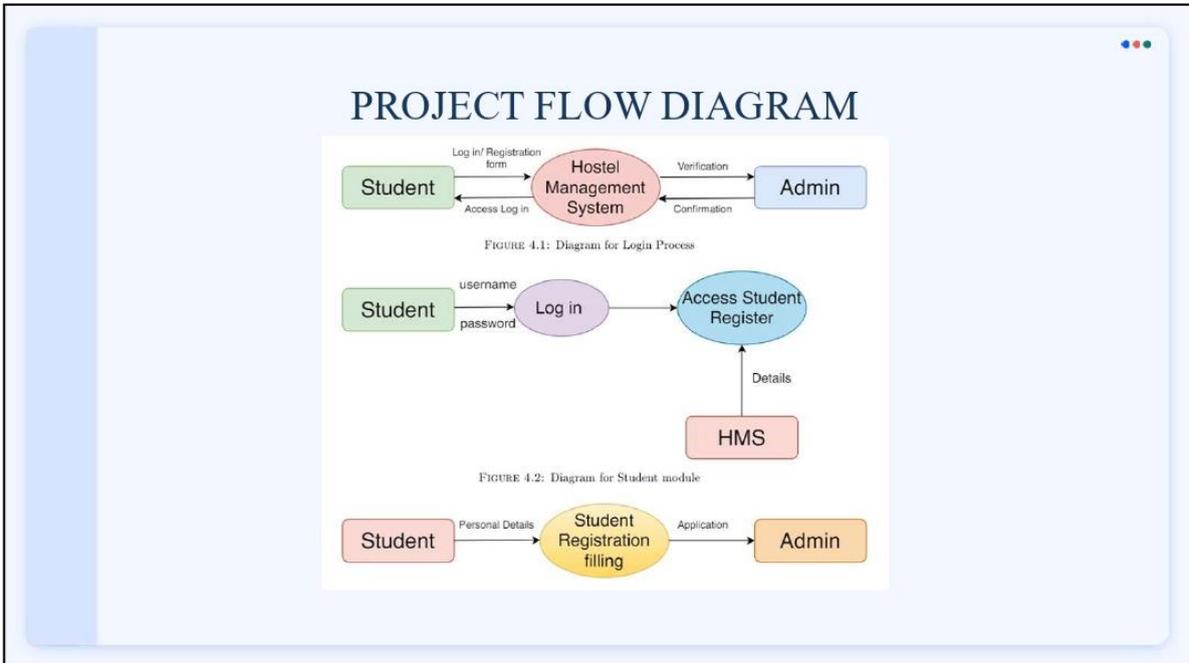
25

FLOW OF PROJECT IN SEM-II

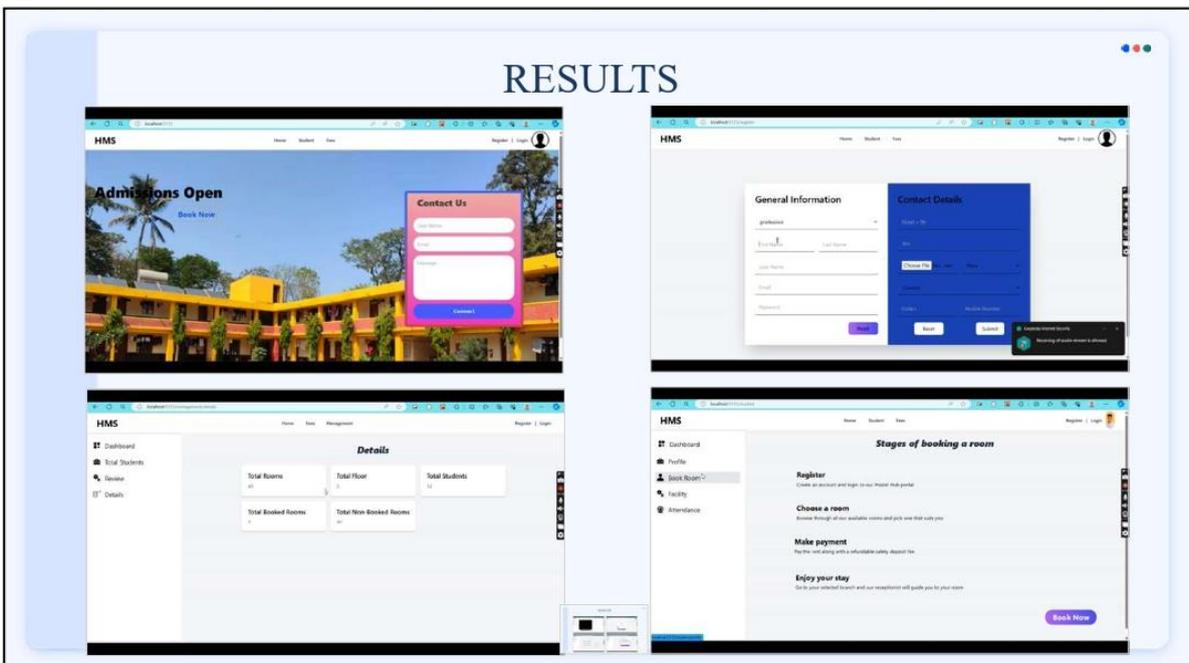
- Project Initiation
- Requirements Gathering and Analysis
- System Design
- Development and Implementation
- Testing
- Deployment and Training
- Maintenance and Support

26

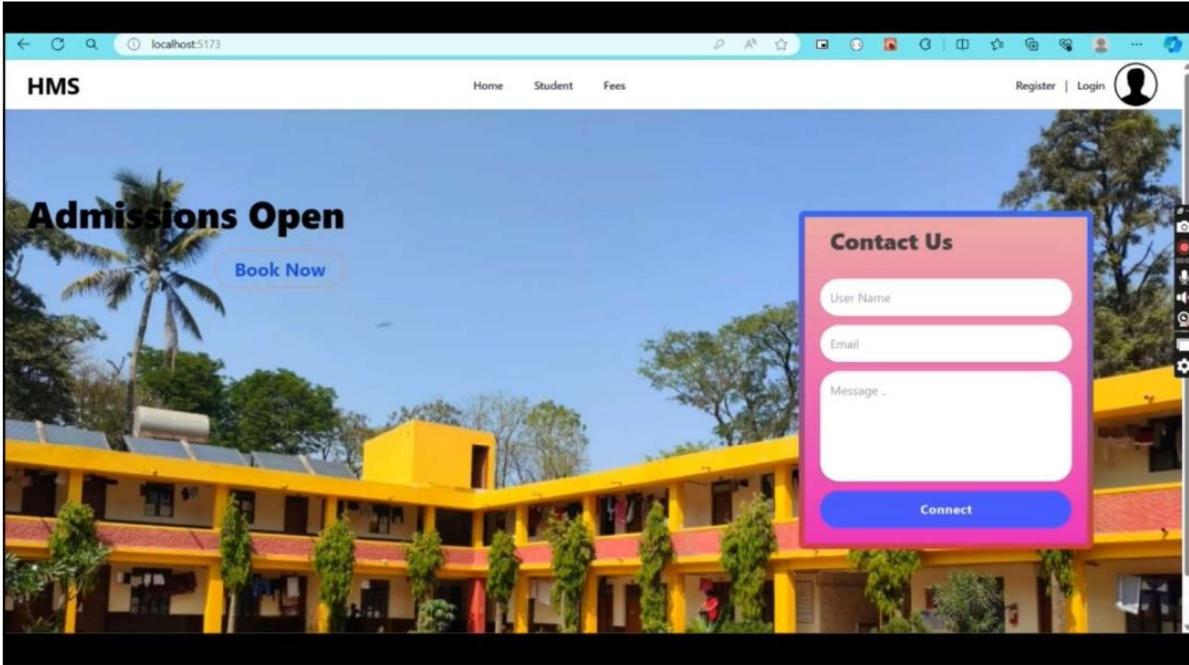
13



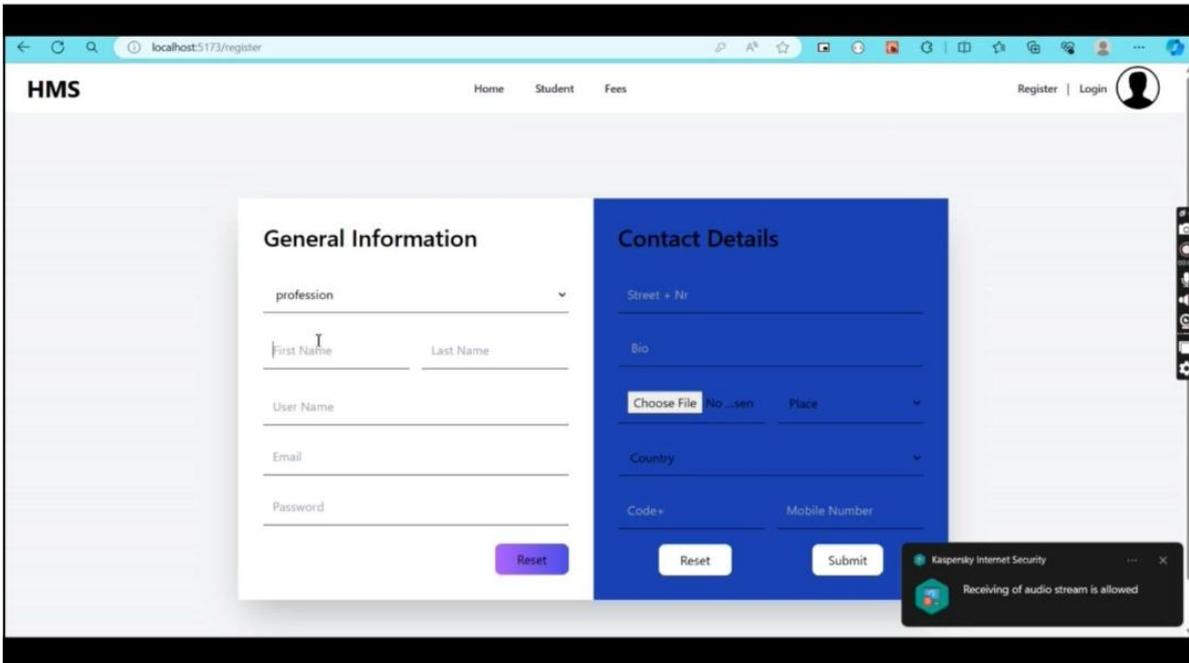
27



28

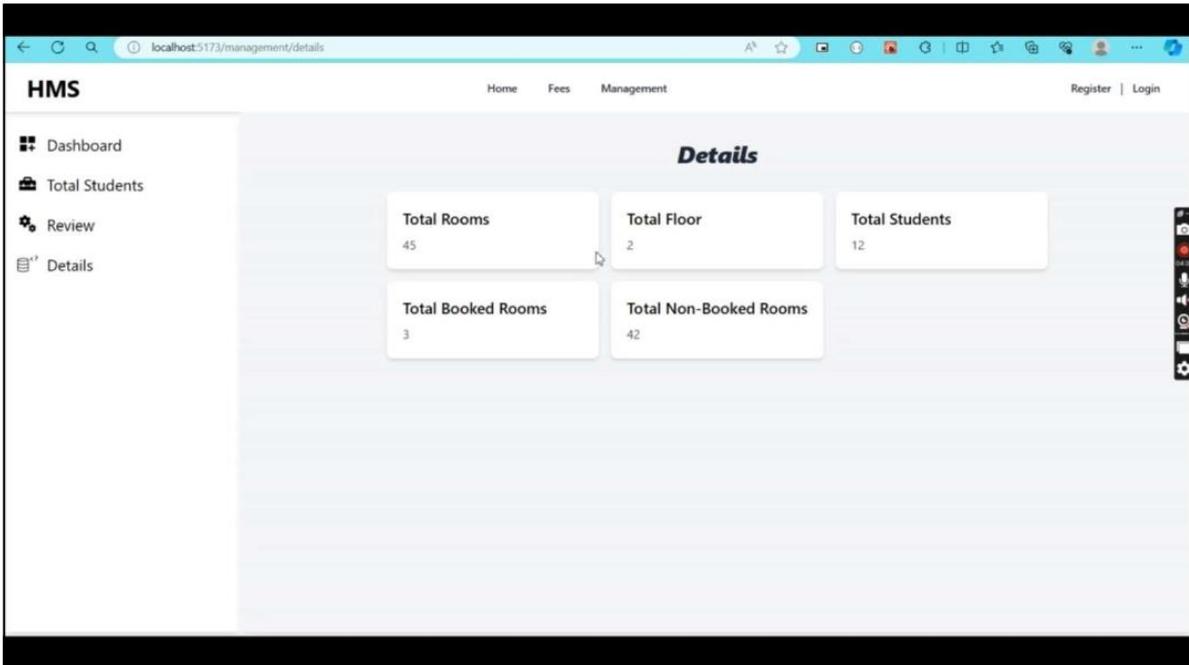


29

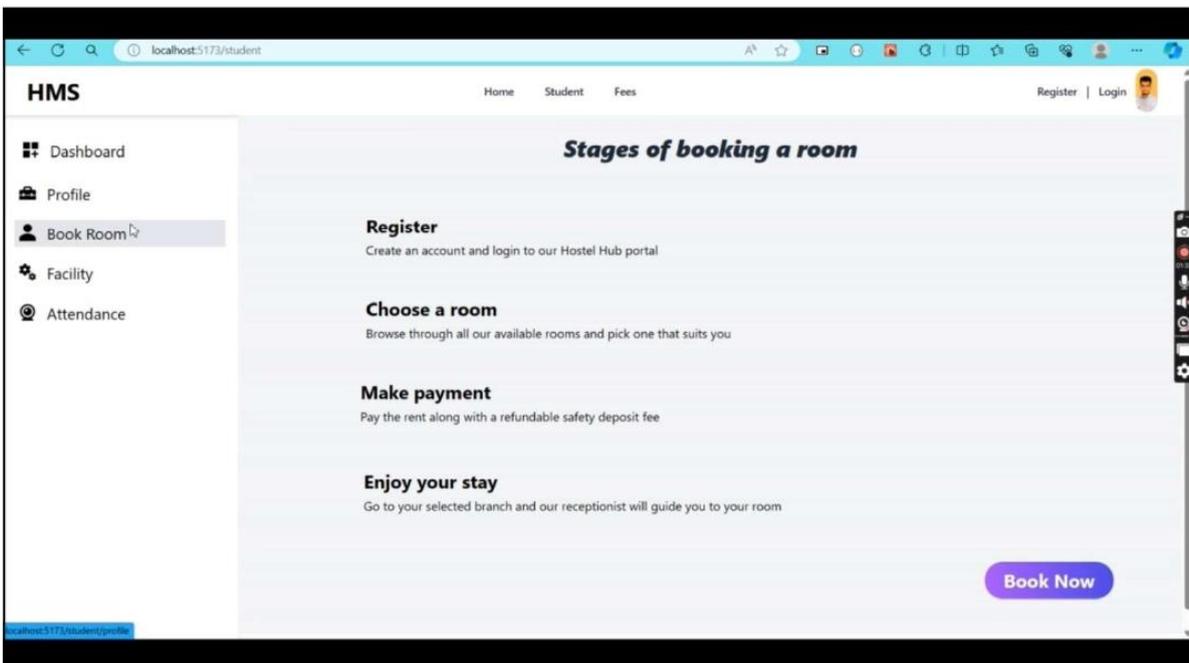


30

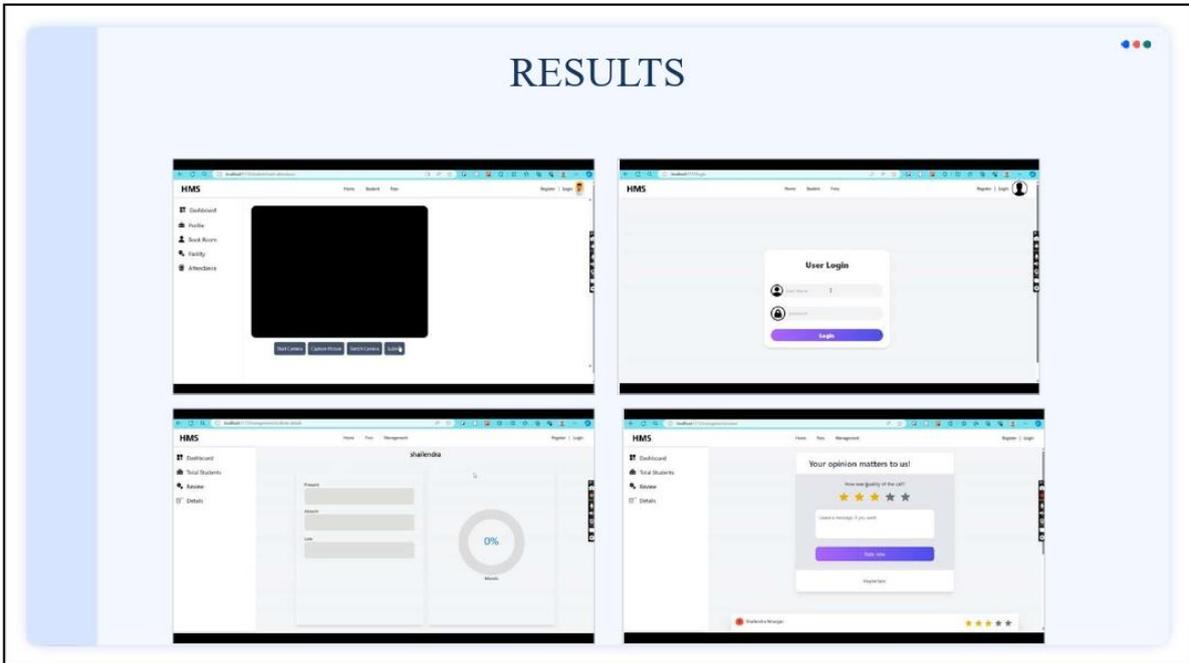
15



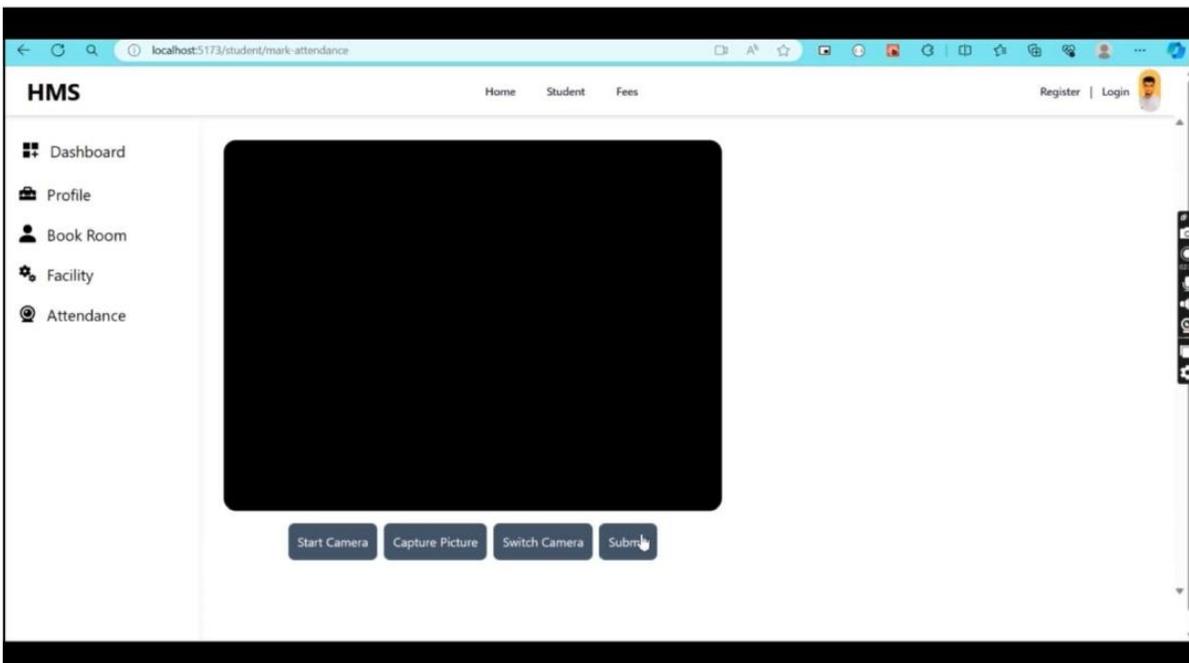
31



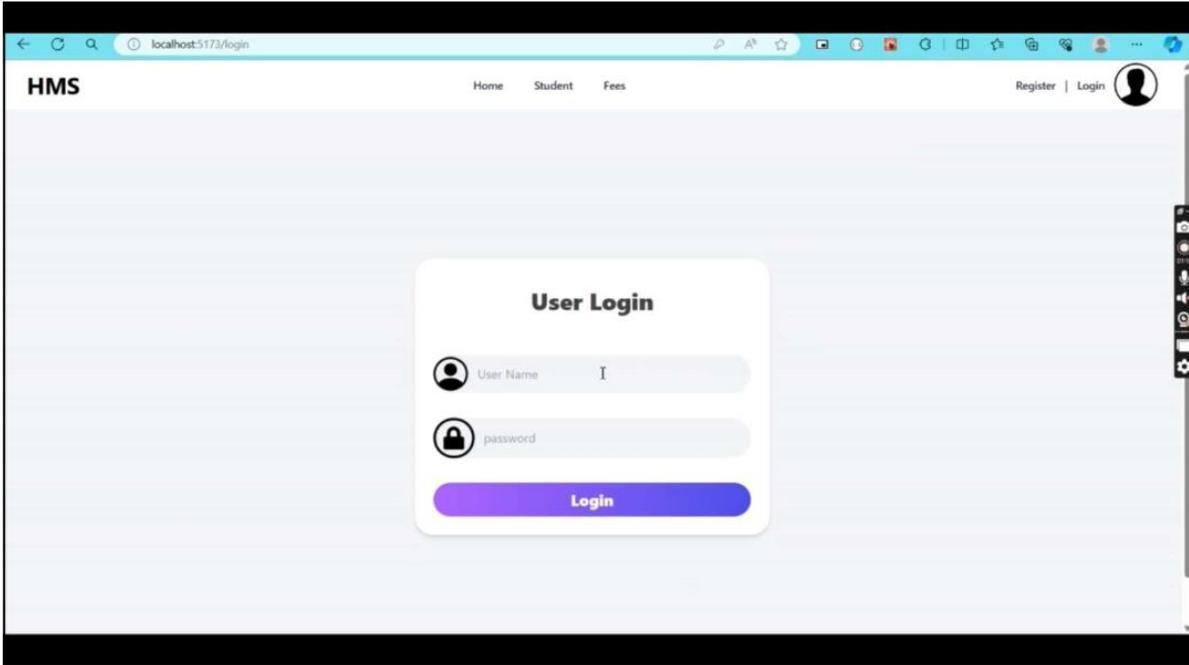
32



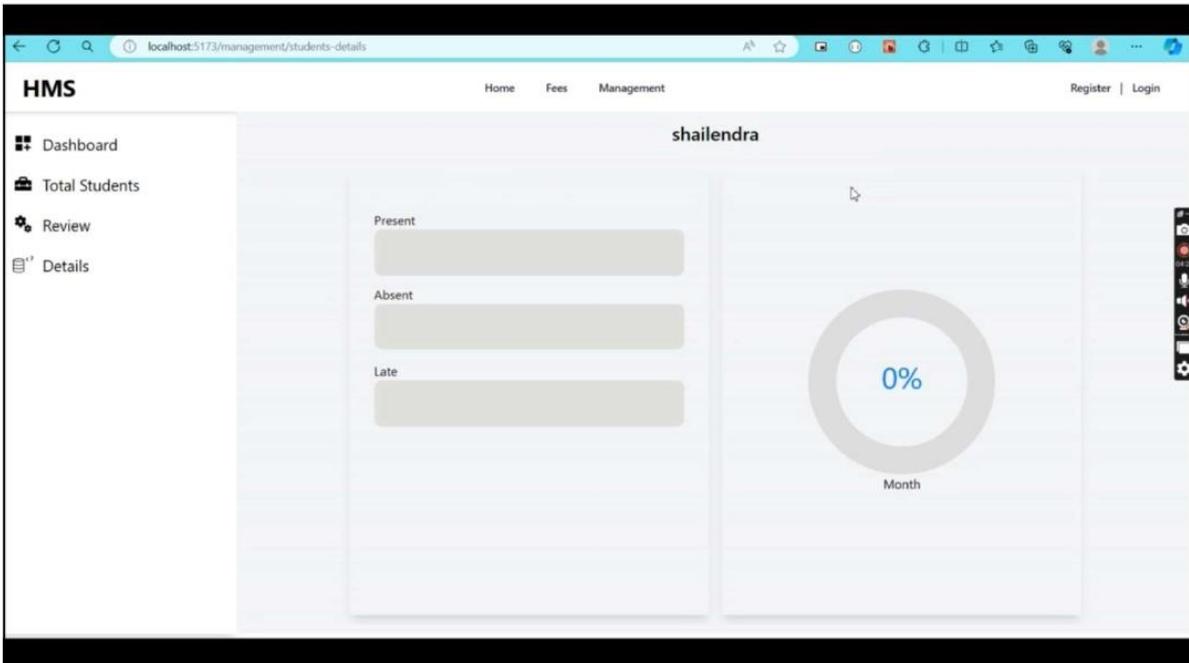
33



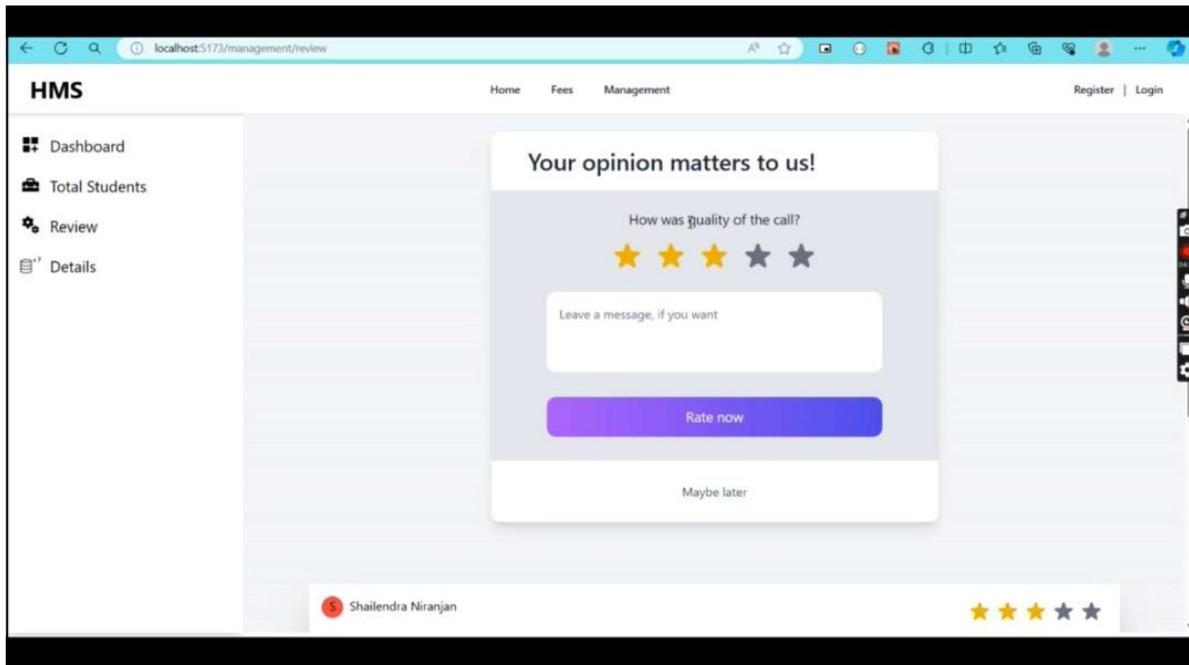
34



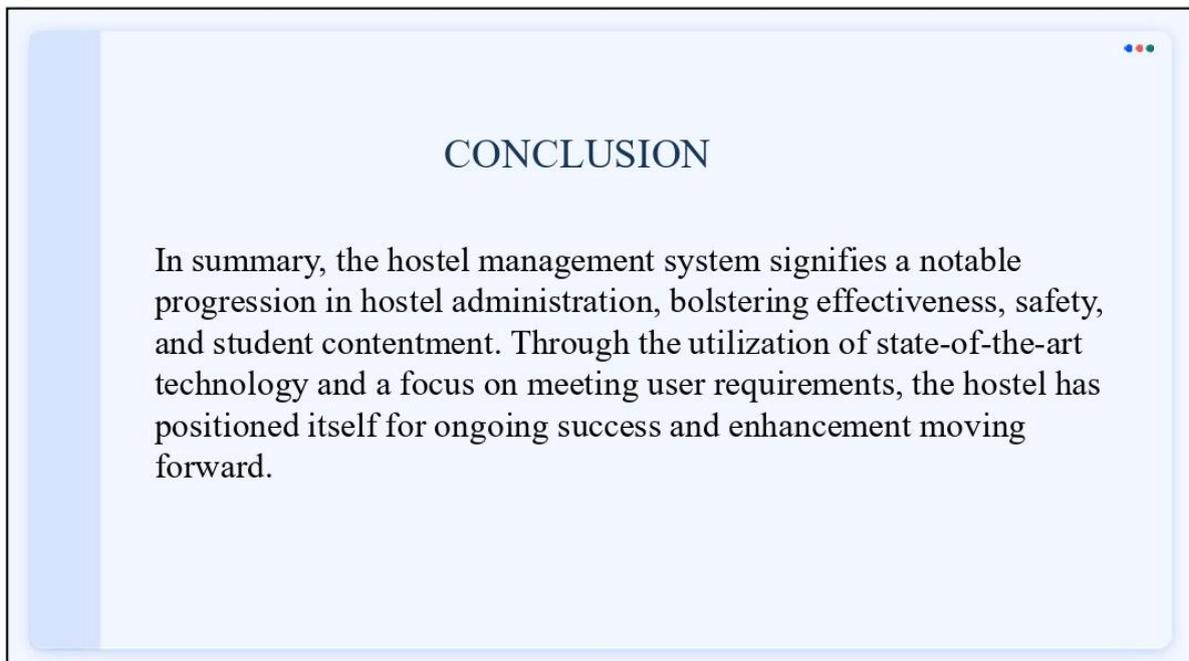
35



36



37



38

19

REFERENCES

- A study on shift towards digitization of Hostel Room Allotment for a university Author Tanmay Nandanwar and Priyanka Bahutule (2022).
- Development of an Automated Hostel Facility Management System Journal of Science and Engineering Askar, Abhishek & Gaur, Saumya & Deollikar, Sejal & Ubale (2021).
- Hostel Management System and Aggregation Online Hostel Management International Journal of Advanced Engineering & Science Research Prof. Shyamsundar Magar, Ms. Sakshi Said and Mr. Rohit Jadhav (2021).
- College Hostel Inventory Management System Sreerambabu, Dr Riyaz, Mr Rajkumar (2023).
- Hostel Management System (HMS) Narkhede, Prof & Bamgude, Rutuja & Sonawane, Mayuri & Shevade, Mandar (2022).
- Hostel Management System Based on Finger Print Authentication ISSN RAJKUMAR, G. & SUNDARI (2018).
- Implementation Of Hostel Management with
- Automation Using Design Thinking the Research of Development Framework Based on MVC Pattern Dinesh.B, Gogul Nithin.R, Pavatharani.R,Sneha.R,C.Senthilkumar (2022).
- Frameworks usability in Web Application Development Wei Cui,Lin Huang and Lüing Liang (2018).

39

THANK YOU

40

20

ANNEXTURE G
PROJECT ACHIEVEMENTS



SAVITRIBAI PHULE PUNE UNIVERSITY

Internal Quality Assurance Cell (IQAC)

AAVISHKAR - 2023

ZONAL LEVEL RESEARCH PROJECT COMPETITION

Certificate of Participation

This is to certify that Mr./Miss. SUJAL THAKARE
of Anandao Pawar COE & Research College/Institute has participated
in Zonal Level Research for UG / PG / Post PG (M.Phil. / Ph.D.) at **"AAVISHKAR 2023"** Research
Festival, Under ENGINEERING & TECH category, held at AISSMS COE
PUNE during 27-10 2023.


Coordinator


Principal


Prof. Sanjay Dhole
Director, IQAC



SAVITRIBAI PHULE PUNE UNIVERSITY

Internal Quality Assurance Cell (IQAC)

AAVISHKAR - 2023

ZONAL LEVEL RESEARCH PROJECT COMPETITION

Certificate of Participation

This is to certify that Mr./Miss. RACHANA PATIL
of Anandao Pawar COE and Research College/Institute has participated
in Zonal Level Research for UG / PG / Post PG (M.Phil. / Ph.D.) at **"AAVISHKAR 2023"** Research
Festival, Under ENGINEERING & TECH category, held at AISSMS COE
PUNE during 27-10 2023.


Coordinator


Principal


Prof. Sanjay Dhole
Director, IQAC


Marathwada Mitra Mandal's College of Engineering
Accredited by NAAC with 'A++' grade | Best College Award by SPPU

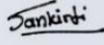

CICS
CULTIVATING IDEAS CRAFTING SOLUTIONS

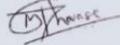
CERTIFICATE
OF APPRECIATION

This certificate is presented to

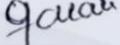
Sujal Suresh Thakre

for participation in CICS IDEATHON conducted by FMCIII in collaboration with ISTE SC AND ACM SC at Marathwada Mitra Mandal College Of Engineering, Pune.


Dr.S.S.Shiravale
Secretary ISTE SC, MMCOE


Mr.N.S.Dhavase
Associate Professor, MMCOE


Dr.V.N.Gohokar
Principal, MMCOE


Dr.C.R. Talathi
C.E.O, FMCIII


Marathwada Mitra Mandal's College of Engineering
Accredited by NAAC with 'A++' grade | Best College Award by SPPU

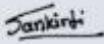

CICS
CULTIVATING IDEAS CRAFTING SOLUTIONS

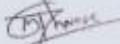
CERTIFICATE
OF APPRECIATION

This certificate is presented to

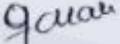
Sadaf Azam Shaikh

for participation in CICS IDEATHON conducted by FMCIII in collaboration with ISTE SC AND ACM SC at Marathwada Mitra Mandal College Of Engineering, Pune.


Dr.S.S.Shiravale
Secretary ISTE SC, MMCOE


Mr.N.S.Dhavase
Associate Professor, MMCOE


Dr.V.N.Gohokar
Principal, MMCOE


Dr.C.R. Talathi
C.E.O, FMCIII



**ANNEXTURE H
DOCUMENTATION OF
EXCECUTION OF
PROJECT**

- **Introduction**

- Overview of the hostel management system project
- Objectives and scope
- Key stakeholders and their roles
- Importance of the camera attendance and automatic room allotment features

- **System Requirements**

- Functional requirements
- Camera attendance (e.g., facial recognition, attendance tracking, reporting)
- Automatic room allotment (e.g., room availability tracking, allocation algorithms, preferences handling)
- Non-functional requirements (e.g., performance, scalability, security)
- Hardware requirements (e.g., cameras, servers, network infrastructure)
- Software requirements (e.g., facial recognition software, attendance management software, room allocation software)

- **System Architecture**

- High-level architecture diagram
- Description of components (e.g., camera subsystem, facial recognition subsystem, attendance management subsystem, room allocation subsystem)
- Data flow and interactions between components

- **Implementation Details**

- Camera attendance implementation
- Camera setup and configurations
- Facial recognition algorithm and libraries used
- Integration with attendance management system
- Automatic room allotment implementation
- Room availability tracking mechanism
- Room allocation algorithm and logic
- Handling of student preferences and priority criteria
- User interface for attendance monitoring, room allocation, and reporting
- Security considerations (e.g., data privacy, access controls)

- **Testing and Quality Assurance**

- Test plan and strategy
- Unit testing, integration testing, and system testing
- Test case documentation for camera attendance and room allocation scenarios
- Accuracy and performance testing of facial recognition and room allocation algorithms

- **Deployment and Operations**

- Camera deployment locations and configurations
- Server and network infrastructure setup
- Data storage and backup procedures
- Monitoring and logging mechanisms

- **User Documentation**

- User manual or guide
- Instructions for students/staff to register facial data and room preferences
- Attendance reporting and room allocation procedures

- **Maintenance and Support**

- Procedures for updating facial recognition models and room allocation algorithms
- Issue tracking and resolution
- Planned enhancements and future roadmap

- **Project Management**

- Project schedule and milestones
- Risk management strategies
- Team collaboration and communication

- **Lessons Learned and Best Practices**

- Challenges faced during implementation (e.g., lighting conditions, occlusions, room allocation constraints)
- Solutions and workarounds implemented
- Best practices and recommendations for future projects

ANNEXTURE I
SYSTEM CODE

Login Page

```
'import React, {useState} from 'react'
import "../Register/Register.css"
import axios from 'axios';
import { useNavigate } from 'react-router-dom';

// import { Link, useNavigate } from "react-router-dom";
import { BsPersonCircle } from "react-icons/bs";
import { FaLock } from "react-icons/fa";
// import Service from "/src/Service/Service.js"
export default function Login() {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const loginUser = () => {
    login({ username, password })
  };
  const url = "http://localhost:8080/api/auth/signin";
  const navigate = useNavigate();
  const login = (data) => {
    axios.post(url , data)
      .then(response => {
        // Handle successful login response
        const userData = response.data;
        // Store the user data in local storage
        localStorage.setItem('user', JSON.stringify(userData));
        // Navigate to the dashboard or any other page after successful login
        // eslint-disable-next-line react-hooks/rules-of-hooks
        navigate('/');
      })
      .catch(error => {
        // Handle login error
        console.error('Login failed:', error);
      });
  };
  return (
```

```

<div className=' bg-gray-100 h-screen flex justify-center items-center '>
  <div className='p-6 transition-all ease-in-out duration-1000 md:w-[30%] w-[70%] bg-
white flex flex-col gap-10 rounded-3xl shadow-md '>
    <div className='text-3xl flex justify-center space-x-3 md:font-extrabold p-3 text-
[#444444]' ><p className="">User Login</p></div>
    <div className='flex flex-col gap-8'>
      <div>
        <div className='flex flex-row rounded-[2rem] items-center bg-gray-100'>
          <BsPersonCircle size={50} /><input type="text" onChange={(e)=>{
            setUsername(e.target.value);
            console.log(username)
          }} className='outline-none rounded-3xl bg-transparent p-3 w-[100%]'
placeholder='User Name' name="" id="" />
        </div>
      </div>
    </div>
    <div>
      <div className='flex flex-row rounded-[2rem] items-center bg-gray-100'>
        <div className='p-2 rounded-full border-[3.5px] border-black'><FaLock
size={30} /></div><input type="password" onChange={(e)=>{
          setPassword(e.target.value);
          console.log(password)
        }} className='outline-none rounded-3xl bg-transparent p-3 w-[100%]'
placeholder='password ' name="" id="" />
      </div>
    </div>
    <div className='flex flex-row justify-center'>
      <input className='p-2 w-[100%] text-white space-x-2 bg-gradient-to-r from-
purple-500 to-indigo-600 text-xl font-extrabold rounded-3xl' onClick={loginUser}
type="button" value="Login" />
    </div>
  </div>
</div>

```

Dashboard

```
import React from 'react'
import { CircularProgressbar } from 'react-circular-progressbar';
import 'react-circular-progressbar/dist/styles.css';
import ProgressBar from "@ramonak/react-progress-bar";
import { FcSearch } from "react-icons/fc";
import './Dashboard.css'
export default function Dashboard() {
  const data = [
    { name: 'John', Monday: 8, Tuesday: 7, Wednesday: 6, Thursday: 8, Friday: 9, Saturday:
5, Sunday: 7 },
    { name: 'Alice', Monday: 7, Tuesday: 6, Wednesday: 8, Thursday: 7, Friday: 6,
Saturday: 7, Sunday: 6 },
    { name: 'Bob', Monday: 8, Tuesday: 8, Wednesday: 7, Thursday: 9, Friday: 8, Saturday:
7, Sunday: 8 },
    { name: 'Emily', Monday: 6, Tuesday: 7, Wednesday: 8, Thursday: 7, Friday: 6,
Saturday: 8, Sunday: 7 },
    { name: 'Michael', Monday: 9, Tuesday: 8, Wednesday: 7, Thursday: 8, Friday: 9,
Saturday: 8, Sunday: 8 },
    { name: 'Sophia', Monday: 8, Tuesday: 9, Wednesday: 8, Thursday: 9, Friday: 8,
Saturday: 7, Sunday: 8 },
    { name: 'David', Monday: 7, Tuesday: 8, Wednesday: 7, Thursday: 8, Friday: 7,
Saturday: 6, Sunday: 7 },
    { name: 'Emma', Monday: 8, Tuesday: 7, Wednesday: 8, Thursday: 7, Friday: 8,
Saturday: 9, Sunday: 8 },
    { name: 'Daniel', Monday: 7, Tuesday: 6, Wednesday: 7, Thursday: 6, Friday: 7,
Saturday: 8, Sunday: 6 },
    { name: 'Olivia', Monday: 8, Tuesday: 9, Wednesday: 8, Thursday: 9, Friday: 8,
Saturday: 7, Sunday: 8 },
    { name: 'William', Monday: 7, Tuesday: 8, Wednesday: 7, Thursday: 8, Friday: 7,
Saturday: 6, Sunday: 7 },
    { name: 'Ava', Monday: 8, Tuesday: 7, Wednesday: 8, Thursday: 7, Friday: 8, Saturday:
9, Sunday: 8 },
    { name: 'James', Monday: 7, Tuesday: 6, Wednesday: 7, Thursday: 6, Friday: 7,
Saturday: 8, Sunday: 6 },
```

```

    { name: 'Mia', Monday: 8, Tuesday: 9, Wednesday: 8, Thursday: 9, Friday: 8, Saturday:
7, Sunday: 8 },
    { name: 'Benjamin', Monday: 7, Tuesday: 8, Wednesday: 7, Thursday: 8, Friday: 7,
Saturday: 6, Sunday: 7 },
    // Add more data as needed
];
const room = [
    { roomID: 101, students: [
        { id: 1, present: true },
        { id: 2, present: false },
        { id: 3, present: true },
        { id: 4, present: false }
    ] },
    { roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },
    { roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },

```

```

        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] },{ roomID: 102, students: [
        { id: 5, present: true },
        { id: 6, present: true },
        { id: 7, present: false },
        { id: 8, present: true }
    ] }
    // Add more room data as needed
];
return (
    <div className='float-right flex-col gap-4 flex bg-gray-100 overflow-y-auto p-3
overflow-x-hidden h-[42.2rem] w-[80%]'>
        <div className={'md:h-[20rem] flex md:flex-row flex-col p-3 bg-white shadow-lg'}>
            <div className={'md:w-[50%] md:h-[100%] h-[50%] w-[100%] mb-3 p-3 shadow-

```

```

lg overflow-y-auto overflow-x-auto '}>
  <tr className={'p-2 bg-slate-200 '}>
    <th className={'p-2 '}>Name</th>
    <th className={'p-2 '}>Monday</th>
    <th className={'p-2 '}>Tuesday</th>
    <th className={'p-2 '}>Wednesday</th>
    <th className={'p-2 '}>Thursday</th>
    <th className={'p-2 '}>Friday</th>
    <th className={'p-2 '}>Saturday</th>
    <th className={'p-2 '}>Sunday</th>
  </tr>

  {data.map((row, index) => (
    <tr className={'p-2 ' } key={index}>
      <td className={'p-2 '}>{row.name}</td>
      <td className={'p-2 '}>{row.Monday}</td>
      <td className={'p-2 '}>{row.Tuesday}</td>
      <td className={'p-2 '}>{row.Wednesday}</td>
      <td className={'p-2 '}>{row.Thursday}</td>
      <td className={'p-2 '}>{row.Friday}</td>
      <td className={'p-2 '}>{row.Saturday}</td>
      <td className={'p-2 '}>{row.Sunday}</td>
    </tr>
  )))
</div>
<div className={'md:w-[50%] w-[100%] md:h-[100%] h-[50%] mb-3 flex flex-col
justify-center items-center '}>
  <div className={'` h-[100%] mx-auto p-3 shadow-lg overflow-y-auto overflow-
x-auto `'}>
    <tr className={'p-2 bg-slate-200 '}>
      <th className={'p-2 '}>Room ID</th>
      <th className={'p-2 '}>Student 1</th>
      <th className={'p-2 '}>Student 2</th>
      <th className={'p-2 '}>Student 3</th>
      <th className={'p-2 '}>Student 4</th>

```

```

</tr>

{room.map((row, index) => (
  <tr className={'p-2 '} key={index}>
    <td className={'p-2 '}>{row.roomID}</td>
    {row.students.map((student) => (
      <td className={'p-2 '}>
        <td className={'p-2 '}>{student.id}</td>
        <td className={'p-2 '}>{student.present ? 'P' : 'A'}</td>
      </td>
    ))}
  </tr>
)}}
</div>

</div>
</div>
<div className={`md:h-[20rem] p-3 bg-white flex md:flex-row flex-col`} >
  <div className={`md:w-[50%] h-[50%] md:h-[100%] w-[100%] p-8 shadow-lg`} >
    <div className={'flex flex-row justify-center items-center '} >
      {/*<div className={`bg-slate-200 rounded-3xl flex flex-row w-[50%] justify-
center items-center p-2`} >*/}
      {/* <input type={"text"} className={'outline-none bg-slate-200 w-[100%] '}
value="" placeholder={"student name"} />*/}
      {/* <FcSearch size={30} />*/}
      {/*</div>*/}
    </div>
    <div className={`flex flex-col gap-4`} >
      <div>
        <span>Present</span> <ProgressBar completed={60} />
      </div>
      <div>
        <span>Absent</span> <ProgressBar completed={60} />
      </div>
    </div>
  </div>

```

```

        <span>Late</span>    <ProgressBar completed={60} />
    </div>
</div>

</div>
<div className={`md:w-[50%] w-[100%] p-3 h-[50%] md:h-[100%] flex flex-col
justify-center items-center shadow-lg`} >
    <div className={`h-[1r5em] w-[15rem]`} >
        <CircularProgressbar
            value={70}
            text={`$${70}%`}
            strokeWidth={15}
            styles={{
                // Customize the root svg element
                root: {},
                // Customize the path, i.e. the "completed progress"
                path: {
                    stroke: '#007bff',
                    strokeLinecap: 'round',
                    transition: 'stroke-dashoffset 0.5s ease 0s',
                },
                // Customize the text
                text: {
                    fill: '#007bff',
                    fontSize: '16px',
                },
                // Customize background - this is the "empty progress"
                trail: {
                    stroke: '#ddd',
                },
            }}
        />
    </div>
<div>
    Month

```

```

        </div>
    </div>
</div>
</div>
)
}

```

Image recognition

```

import sys
import cv2
import face_recognition
import numpy as np
import pandas as pd

enc_csv = pd.read_csv(r"C:\Users\Lenovo\OneDrive\Desktop\final sub\hms-backend\hms-
backend\harvest-backend\src\main\java\com\hmsbackend\scripts\encodings.csv")
encodings_to_list = [enc.split(' ') for enc in enc_csv['encodings']]
known_face_encodings = [[float(encoding) for encoding in i] for i in encodings_to_list]
known_face_names = list(enc_csv['name'])
process_this_frame = True

def recognise(img_path):
    img = cv2.imread(img_path)
    frame = cv2.flip(img, 1)
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
    # rgb_small_frame = small_frame[:, :, ::-1]
    rgb_small_frame = img
    if process_this_frame:
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(
            rgb_small_frame, face_locations)
        face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(known_face_encodings,

```

```

        face_encoding)

    name = ""
    face_distances = face_recognition.face_distance(
        known_face_encodings, face_encoding)
    best_match_index = np.argmin(face_distances)
    if matches[best_match_index]:
        name = known_face_names[best_match_index]
    face_names.append(name)
    return face_names[0]

def main():
    print(recognise(sys.argv[1]))

if __name__ == "__main__":
    main()

```

Backend

```

package com.hmsbackend;

import com.hmsbackend.model.*;
import com.hmsbackend.repository.RoleRepository;
import com.hmsbackend.repository.RoomRepository;
import com.hmsbackend.repository.UserRepository;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.security.crypto.password.PasswordEncoder;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

@SpringBootApplication
public class HarvestBackendApplication {

```

```

public static void main(String[] args) {
    SpringApplication.run(HarvestBackendApplication.class, args);
}

@Bean
CommandLineRunner run(RoleRepository roleRepository , UserRepository
userRepository , PasswordEncoder passwordEncoder , RoomRepository roomRepository){

    return args -> {
        Role adminRole = null;

        if(roleRepository.findByName(ERole.ROLE_ADMIN).isPresent())return;
        adminRole = roleRepository.save(new Role(1 ,
ERole.ROLE_ADMIN));
        roleRepository.save(new Role(2 , ERole.ROLE_USER));

        Set<Role> roles = new HashSet<>();
        roles.add(adminRole);

        User admin = new User("admin771@gmail.com" ,"admin",
passwordEncoder.encode("admin"),roles);
        admin.setFirstName("admin");
        admin.setLastName("admin");
        userRepository.save(admin);
        for(int i=1; i<=25; i++){
            Room room = new Room();
            room.setDate(new Date());
            room.setRoomNo(i);
            room.setFee(5100);
            room.setFloor(Floor.GROUND);
            room.setNumberOfBeds(4);
            room.setTotalStudents(0);
            roomRepository.save(room);

        }
    }
}

```

```
        for(int i=26; i<=45; i++){
            Room room = new Room();
            room.setDate(new Date());
            room.setRoomNo(i);
            room.setFee(5100);
            room.setFloor(Floor.FIRST);
            room.setNumberOfBeds(3);
            room.setTotalStudents(0);
            roomRepository.save(room);
        }
    };
}
};
}
```