Hostel Management System

A PROJECT REPORT

submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

in

COMPUTER ENGINEERING

Major Project I (01CE0716)

Submitted by

Dhairya Aundhia

92200103268

Kunj Nirmal

92200103285

Rishi Patel

92200103288

Faculty of Engineering & Technology

Marwadi University, Rajkot

August, 2025

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Hostel Management System** has been carried out by **Dhairya Aundhia** (92200103268) **Kunj Nirmal** (92200103285) **Rishi Patel** (92200103288) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Parth Shah	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Hostel Management System** has been carried out by **Dhairya Aundhia 92200103268** under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Parth Shah	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Hostel Management System** has been carried out by **Kunj Nirmal 92200103285** under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Parth Shah	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Hostel Management System** has been carried out by **Rishi Patel 92200103288** under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Parth Shah	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Major Project (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

DECLARATION

We hereby declare that the **Major Project-I (01CE0716)** report submitted along with the Project entitled **Online Lost and Found Portal For College Campus** submitted in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering to Marwadi University, Rajkot, is a bonafide record of original project work carried out by me / us at Marwadi University under the supervision of **Prof. Khanjan Trivedi** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

S.No	Student Name	Sign
1	Dhairya Aundia	
2	Kunj Nirmal	
3	Rishi Patel	

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of my project, ***Hostel Management System***

First and foremost, I am deeply thankful to Prof. Khanjan Trivedi, my project guide, for their continuous guidance, valuable suggestions, and encouragement throughout the development of this project.

I am also grateful to the Department **of Computer Engineering Marwadi University** for providing the necessary facilities, resources, and a supportive environment to carry out this work.

I would like to extend my appreciation to my classmates, friends, and family members for their constant support, motivation, and constructive feedback.

Finally, I express my heartfelt thanks to all who directly or indirectly helped me in making this project a success.

Abstract

The Hostel Management System (HMS) is a web-based platform that centralizes and automates core hostel operations—room and bed allocation, attendance tracking, fee management, maintenance requests, mess management, and reporting—for educational institutions to improve service quality and reduce manual effort. Built on the MERN stack with role-based access control, the system delivers a responsive UI, secure authentication, and real-time dashboards for wardens and administrators, while providing self-service capabilities for students. By replacing paper-driven workflows with auditable, structured records and modular services, HMS lowers errors, speeds approvals, and enables data-driven decisions across the hostel lifecycle

List of Figures

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Project Overview

A hostel is a critical infrastructure component for educational institutions, providing safe and organized accommodation for students who relocate from different regions.
As enrolments grow, manual workflows—paper registers for student data, room allocation, fee tracking, attendance, and complaint handling—become slow, error-prone, and difficult to audit.
To address these challenges, the proposed Hostel Management System (HMS) is a centralized, web-based platform that digitizes and automates end-to-end hostel operations, improving accuracy, transparency, and turnaround time for both administrators and students.

- **What HMS Enables:**

The system unifies core functions—student records, room and bed allocation, attendance capture, fee invoicing and tracking, maintenance/complaint workflow, mess-off requests, and parcel logging—under a single interface with role-based access.
Students can securely log in to view allocation details, submit complaints, request mess-off days, check parcels, and track request status, while wardens/admins allocate rooms, monitor attendance, process fees, act on tickets, and generate reports.
All transactions are stored in a centralized database with audit trails, reducing redundancy and enabling quick retrieval, analytics, and policy compliance.

- **Implementation Notes:**

The solution is implemented on the MERN stack—MongoDB, Express, React, and Node.js—providing a responsive UI, RESTful APIs, secure authentication/authorization, and scalable data models.
Dashboards and notifications surface pending allocations, dues, and maintenance SLAs, while structured records and reports support data-driven decision-making by hostel authorities.
If a PHP/MySQL stack is required for deployment constraints, the same functional scope can be retained by substituting the persistence and API layers without altering user roles or process flows.

- **Impact:**

By replacing fragmented paper workflows with a unified digital system, HMS minimizes manual effort and errors, shortens approval cycles, and enhances transparency for students and staff.
Institutions benefit from faster service delivery, improved accountability, and reliable reporting, ultimately elevating the overall hostel experience and operational efficiency.

1.2 Background

Hostel facilities are essential to universities and colleges, but rising enrollments have exposed limits of paper-based processes used for student registration, room allocation, fee tracking, complaints&attendance.
Manual registers and ad-hoc spreadsheets are time-consuming, error-prone, hard to audit, and offer little transparency to students or wardens, leading to delays in allocation, fee reconciliation, resolution.
A modern, centralized web system addresses these gaps by standardizing data capture, enforcing role-based workflows for Students, Wardens, and Admins, and maintaining an auditable record. By moving from fragmented paperwork to a responsive, MERN-based platform consistent with this project, institutions can reduce errors, shorten cycle times, strengthen accountability, and provide students with self-service access and timely updates.

1.3 Need for the System

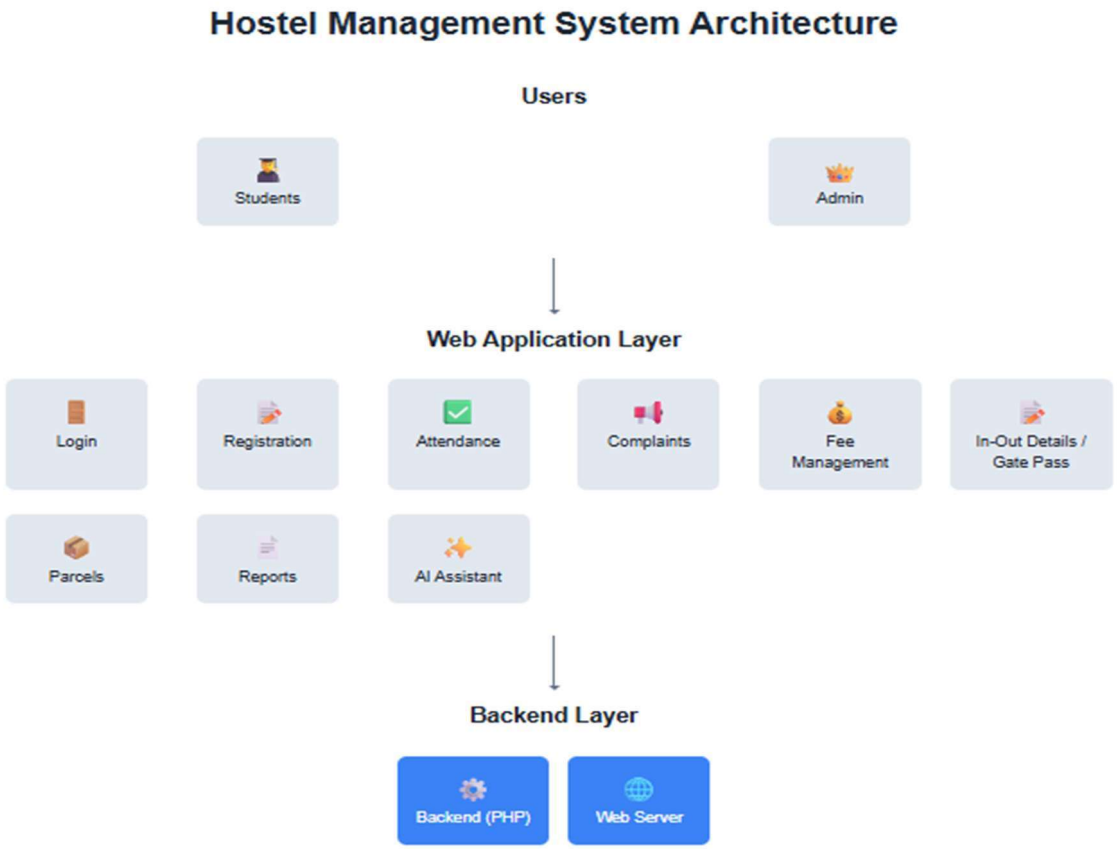
- Manual processes cause duplicated records, slow lookups, and inconsistent updates across departments, creating delays and audit gaps in hostel operations.
- A centralized, web-based HMS is required to consolidate data and standardize workflows for Students, Wardens, and Admins with role-based access.
- The system must enable quick retrieval of student details, allocations, fees, attendance, and complaints, reducing turnaround times for common tasks.
- Built-in tracking for complaints, maintenance requests, and approvals increases transparency and accountability throughout the process lifecycle.
- Accurate, single-source records minimize errors and redundancy while improving compliance and auditability through complete histories and reports.
- Responsive dashboards and notifications surface pending dues, open tickets, and allocation status to support timely decisions and actions.
- Self-service features for students (view allocation, fees, requests) reduce counter traffic and free staff for higher-value tasks.

1.4 Objectives of the Project

- Automate student registration and room/bed allocation with approvals.
- Maintain validated student profiles, attendance, and fee status in one source.
- Track complaints/maintenance with status updates and timely resolution.
- Manage mess-off requests and visitor logs with audit trails.
- Handle fee invoicing, receipts, and hostel expenses as configured.
- Generate configurable reports and real-time dashboards for decisions.
- Strengthen communication via notifications and self-service portal.
- Enforce security with authentication, authorization, and data protection.
- Reduce paperwork and turnaround time through end-to-end digitization**.**

1.5 Scope of the Project

- **Student management**: registration, room/bed allocation, and secure storage of personal and basic academic details.
- **Attendance tracking**: daily in/out logs, leave/mess-off requests, and approvals with audit history.
- **Complaint/maintenance**: ticket creation, status tracking, SLA-based resolution, and closure notes.
- **Fees and expenses:** fee invoicing, receipts, dues tracking, and hostel expense logging with reports.
- **Dashboards and reports**: role-based views for allocations, occupancy, attendance, dues, and complaints.
- **Notifications and communication:** alerts for allocations, dues, approvals, and ticket updates.
- **Security and access:** authentication, authorization, and data integrity for all modules.


Hostel Management System Architecture
Users
Students
Admin
Web Application Layer
Login
Registration
Attendance
Complaints
Fee Management
In-Out Details / Gate Pass
Parcels
Reports
AI Assistant
Backend Layer
Backend (PHP)
Web Server

Chapter 2

Literature Review/Existing System

2.1 Existing System Overview

Most hostels still manage student registration, room allocation, fee collection, complaints, and attendance through paper registers or spreadsheets, which do not scale with rising intake and distributed&stakeholders.

Manual recording of requests—such as mess-off or complaint submissions—introduces delays, data duplication, and limited traceability, often resulting in lost entries.

The absence of a centralized, auditable system reduces transparency for students and wardens and makes timely reporting and decision-making difficult for administration.

2.2 Limitations of the Existing System

The manual or semi-digital system currently used has several drawbacks:

- **Time-consuming operations**: routine registration, allocation, attendance, and complaint handling require manual collation across ledgers/spreadsheets.
- **Error-prone entry:** handwritten and ad-hoc updates introduce mistakes in personal data, allocations, fees, and attendance.
- **Low transparency:** students cannot self-track the status of complaints, mess-off requests, or approvals in real time.
- **Data inconsistency:** multiple registers/files create duplicates and mismatches; there is no single source of truth.
- **Limited accessibility:** records are not centralized, so retrieval and cross-checking are slow and location-bound.

- **Inefficient reporting:** occupancy, dues, and attendance summaries require manual collation and delay decision-making.
- **Weak auditability:** limited end-to-end history of changes, approvals, and resolution times hampers accountability.
- **Security risks:** paper files and shared spreadsheets lack robust authentication, authorization, and data protection.
- **Poor scalability:** processes degrade as student intake and hostel blocks grow, increasing delays and errors.
- **Communication gaps:** reliance on notices and manual follow-ups leads to missed updates and inconsistent information flow.

2.3 Proposed System Overview

The proposed *Hostel Management System* (HMS) is a web-based application designed to overcome the limitations of the existing system. It provides a centralized platform where all hostel operations can be managed efficiently.

- **For Admins:** The system offers modules for student registration, fee management, attendance, staff salary, complaints, and parcel management. Admins can easily generate reports and monitor hostel activities.
- **For Students:** The system allows students to log in to view their details, register complaints, apply for mess-off, and track parcel deliveries.

Key features include:

- Centralized database for secure data storage
- Automated reports and analytics
- Quick search and retrieval of student records
- User-friendly interface for both admin and students
- Improved efficiency and transparency in hostel operations.

2.4 Feasibility Study

A feasibility study evaluates whether the proposed system is practical and achievable.

- **Technical feasibility**

- Built with the MERN stack: MongoDB, Express, React, Node.js; all widely adopted, open-source, and well documented.
- Deployable on standard on-prem or cloud servers with a browser-based client; no specialized hardware required beyond reliable internet.
- Supports authentication, authorization, REST APIs, and modular services for future features (e.g., QR/biometric, online payments).

- **Operational feasibility**

- Role-based UI for Admins/Wardens/Students is intuitive and reduces manual workload via automated workflows and validations.
- Self-service for students (allocations, fees, mess-off, complaints, parcels) increases transparency and reduces counter traffic.
- Dashboards, notifications, and auditable histories improve monitoring, accountability, and day-to-day coordination.

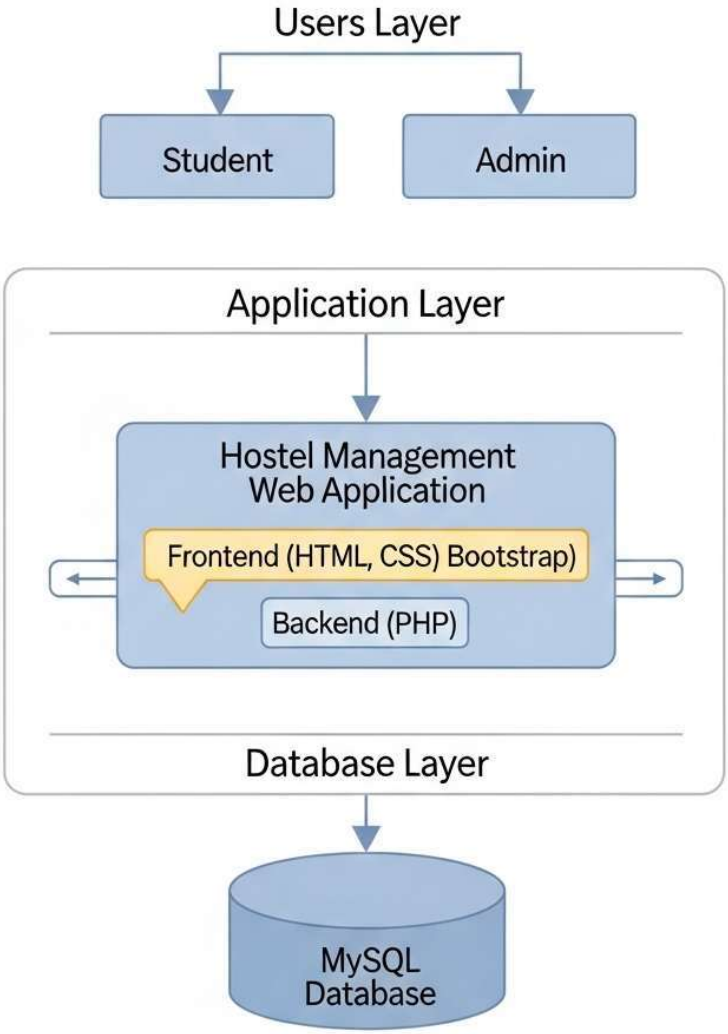
- **Economic feasibility**

- Open-source stack minimizes licensing; hosting and maintenance scale with usage, keeping TCO predictable.
- Paper/process savings plus faster turnarounds reduce operational overheads for staff and administration.
- Reusable components and APIs shorten future enhancement cycles, lowering incremental costs.

- **Time feasibility**

- A semester timeline is practical using agile sprints for core modules: auth, student records, allocation, fees, attendance, complaints.
- Architecture supports phased rollout and later additions (analytics, biometric/QR attendance, online payments) without rework.

2.5 System Architecture Diagram



Chapter 3

System Analysis

3.1 Related Work in Hostel Management Systems

Early web-based HMS solutions (often PHP/MySQL) handled registration and room allocation but struggled with scalability and lacked integration for fees, staff salary, or parcel tracking. Android apps enabled students to lodge complaints and check mess-off schedules but provided limited admin workflows and back-office controls.
ERP/college portals bundled generic hostel modules, yet they were not tailored for hostel operations, causing gaps in attendance capture, visitor/parcel logs, and complaint resolution. Some deployments added QR-based entry/attendance, but remained siloed and did not unify allocation, fees, maintenance, and reporting under one system.

Gap statement

Most solutions are feature-fragmented—covering a single area such as allocation, attendance, or complaints—and lack end-to-end workflows, dashboards, audit trails, and configurable reports; a comprehensive, centralized, role-based HMS is still required.

Alignment note

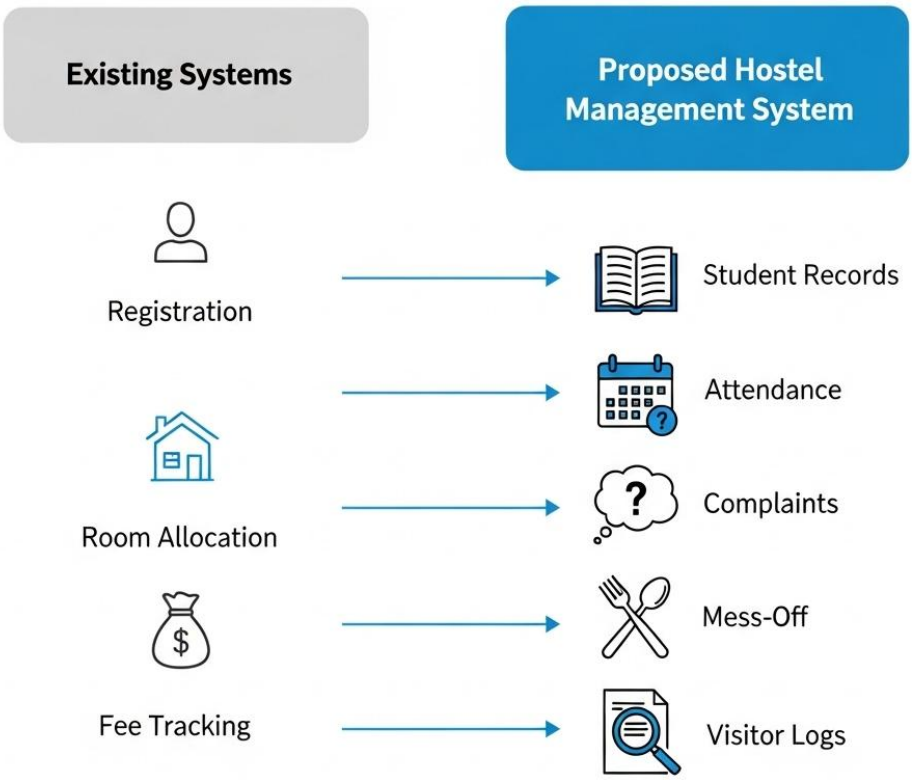
The proposed HMS in this project addresses the above gaps by centralizing core modules with role-based access, real-time dashboards, and auditable records for students, wardens, and admins.

3.2 Summary of Findings

- Existing solutions are often partial, covering only segments of the hostel lifecycle (e.g., allocation or attendance) rather than end-to-end operations.
- Many tools provide basic automation but lack integrated complaints/maintenance, staff/expense tracking, and unified financial records on a single platform.

- A centralized web-based, role-aware system is needed to provide real-time access for students, wardens, and administrators.
- Design must emphasize security, scalability, data accuracy, and auditability to ensure trust and long-term sustainability.

3.3 Conceptual Diagram



Chapter 4

Proposed Approach

4.1 System Design Goals

The proposed Hostel Management System (HMS) aims to overcome the limitations of manual and paper-based management by providing a fully automated, centralized, and secure digital solution.
The main design goals include:

- **Automation:** Replace paper tasks with workflow-driven processes for registration, allocation, attendance, fees, complaints, and mess-off—keeping records current in real time.

- **Accessibility**: Provide secure, web-based access for students and admins from any device with role-appropriate views and actions.

- **Efficiency and accuracy:** Enforce validations, de-duplication, and structured forms to speed entry and retrieval while reducing human error.

- **Security and privacy:** Use authentication/authorization with least-privilege roles, encrypted credentials, and restricted execution of sensitive operations.

- **Scalability and extensibility:** Support multi-hostel growth and new modules (e.g., online payments, biometric/QR attendance) without major redesign.

- **Usability:** Deliver a responsive, modern UI that non-technical users can navigate quickly; framework choice can include Bootstrap or equivalent components.

- **Auditability and transparency:** Maintain histories for requests, approvals, and resolutions with dashboards and notifications for timely follow-ups.

- **Interoperability:** Expose RESTful APIs for integration with institutional systems and exports for analytics/reporting.

- **Reliability and continuity:** Support regular backups and recovery procedures to protect operational data.

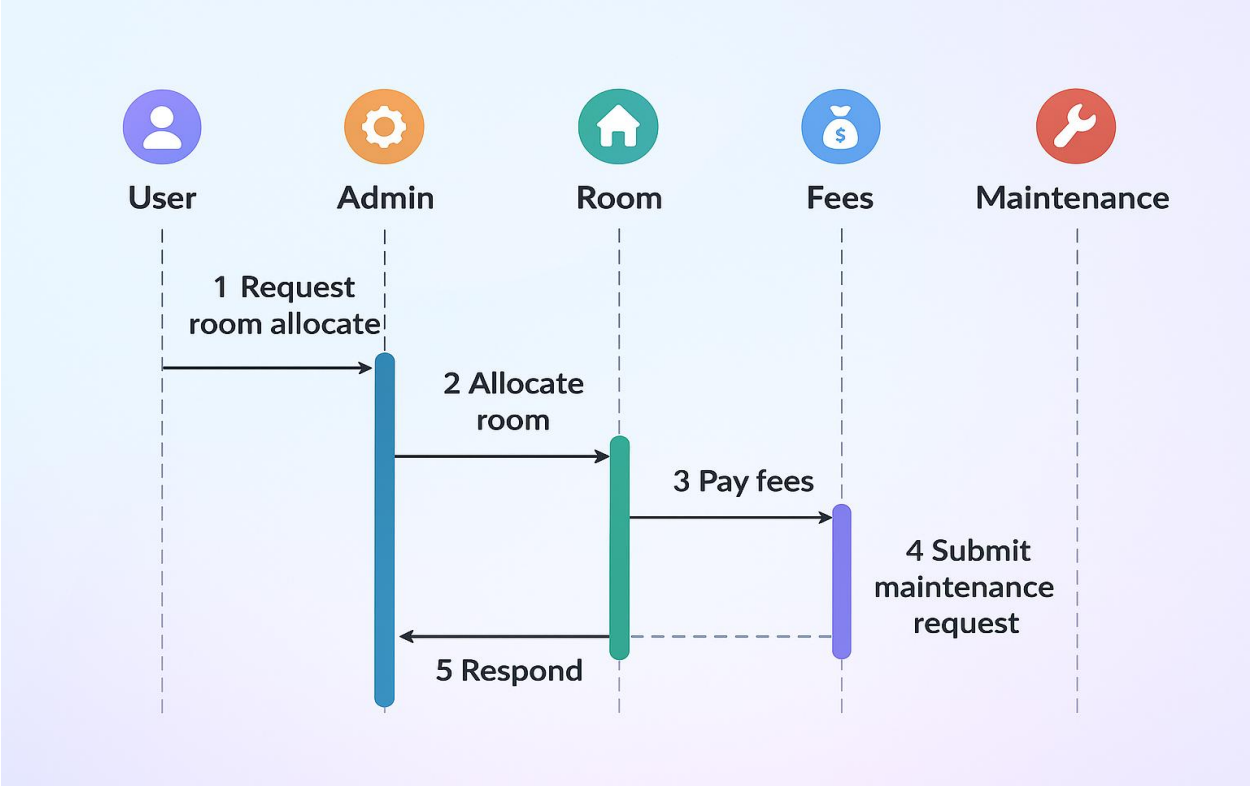
4.2 Key Features

Core modules

- **Student registration & room/bed allocation:** capture personal details, validate capacity, prevent duplicates, and issue credentials for dashboard access.
- **Student dashboard:** show allocation, attendance, fee status, mess-off, complaints, parcels/visitors, and allow limited profile updates.
- **Admin dashboard:** central control with KPIs (occupancy, dues, open tickets, staff cost) and quick links to manage students, fees, staff, maintenance, and reports.
- **Complaint/maintenance management:** students raise tickets (electricity, plumbing, Wi-Fi), admins change status (Pending/In-Progress/Resolved), with timestamps and notifications.
- **Attendance & in-out tracking:** record daily entry/exit and leave requests; enable monitoring and audits for safety and discipline.
- **Mess-off management:** accept applications and adjust billing automatically for approved periods.
- **Fee management:** track dues and payments, generate receipts, and produce finance reports for reconciliation.
- **Staff management:** maintain staff records (wardens, cleaners, cooks, guards) and compute salaries/expenses.
- **Parcel & visitor logs:** record parcels (sender/date) and visitor entries (name/relation/reason/date) to improve transparency and security.
- **Reports & analytics:** export student lists, fee summaries, complaint SLAs, staff salaries, mess-off records, and occupancy trends.

Cross-cutting features

- **Security & roles:** authentication, authorization, and audit trails for all actions.
- **Search & retrieval**: fast, centralized lookups across students, allocations, fees, attendance, and tickets.
- **Notifications:** alerts for allocations, dues, approvals, and ticket updates to reduce turnaround time.



4.3 Technologies Used

- **Frontend:** React + HTML5, CSS3, JavaScript (ES6) — component-based UI, routing, forms, and state management for student/admin dashboards.

- **UI framework:** Bootstrap 5 (or Material UI) — responsive layouts and accessible components for a modern interface.
- **Backend:** Node.js with Express.js — REST APIs, routing, validation, and role-based authorization.
- **Database:** MongoDB (with Mongoose ODM) — stores students, rooms, allocations, fees, attendance, complaints, parcels, and logs.
- **Authentication & security**: JWT, bcrypt hashing, RBAC, Helmet/CORS, HTTPS — secure login/logout and protected resources.
- **Reports & analytics:** MongoDB aggregation + API exports (CSV/PDF) and admin dashboards for fees, attendance, complaints, and occupancy.
- **Server/hosting:** Node runtime on on-prem/cloud (Nginx/Apache reverse proxy optional); MongoDB Atlas or managed MongoDB.
- **Version control:** Git + GitHub/GitLab

Chapter 5

System Design

5.1 System Architecture

- **Presentation layer:** Responsive React UI with HTML5/CSS3/JavaScript and Bootstrap/Material UI for student and admin portals, including forms, tables, dashboards, and notifications.

- **Application layer:** Node.js + Express services implementing authentication/authorization (JWT, RBAC), input validation, session/token handling, and module logic for complaints, attendance, mess-off, fees, parcels, staff, and reports via REST APIs.

- **Data layer:** MongoDB with Mongoose schemas and indexes for students, rooms/allocations, fees, attendance, complaints, parcels, visitors, and staff, enforcing data integrity through schema validation and references.

- **Cross-cutting concerns**

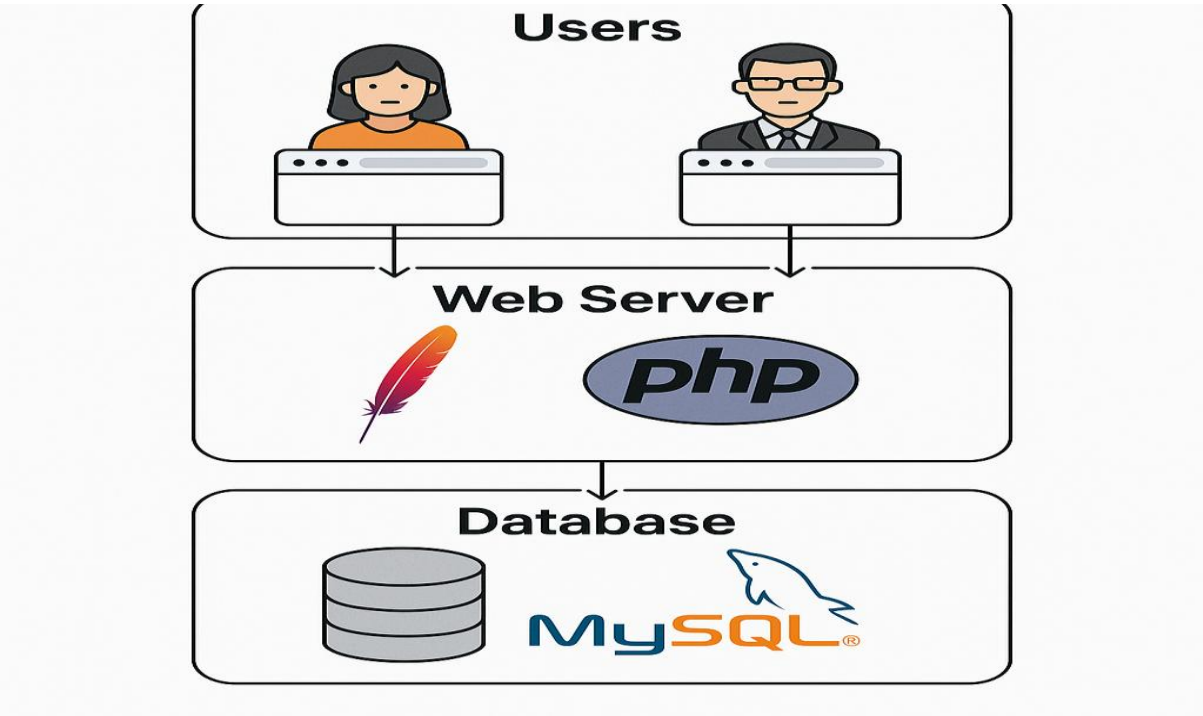
- **Security:** encrypted credentials (bcrypt), role-scoped endpoints, CORS/Helmet, and HTTPS for transport security.

- **Performance & scalability:** stateless APIs, indexed queries, pagination, and horizontal scaling of Node instances and MongoDB collections/replicas.

- **Reliability & auditability:** centralized logs, request/approval histories, and scheduled backups/exports for compliance and recovery.

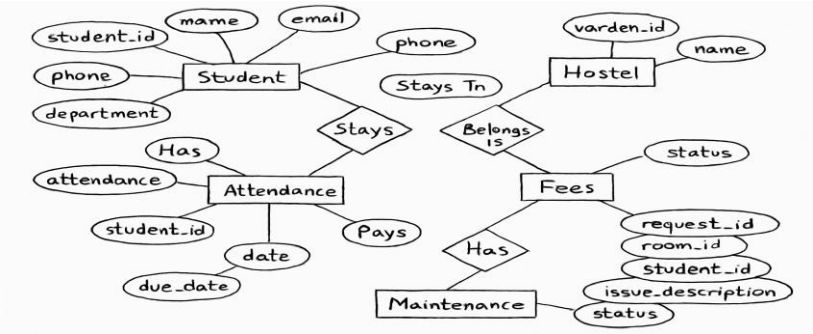
- **Rationale**

This MERN architecture preserves the original three-tier design while improving developer velocity, API integration, and cloud deployment options, and it aligns with the project's stated tech stack.



5.2 Entity-Relationship Diagram (ERD)

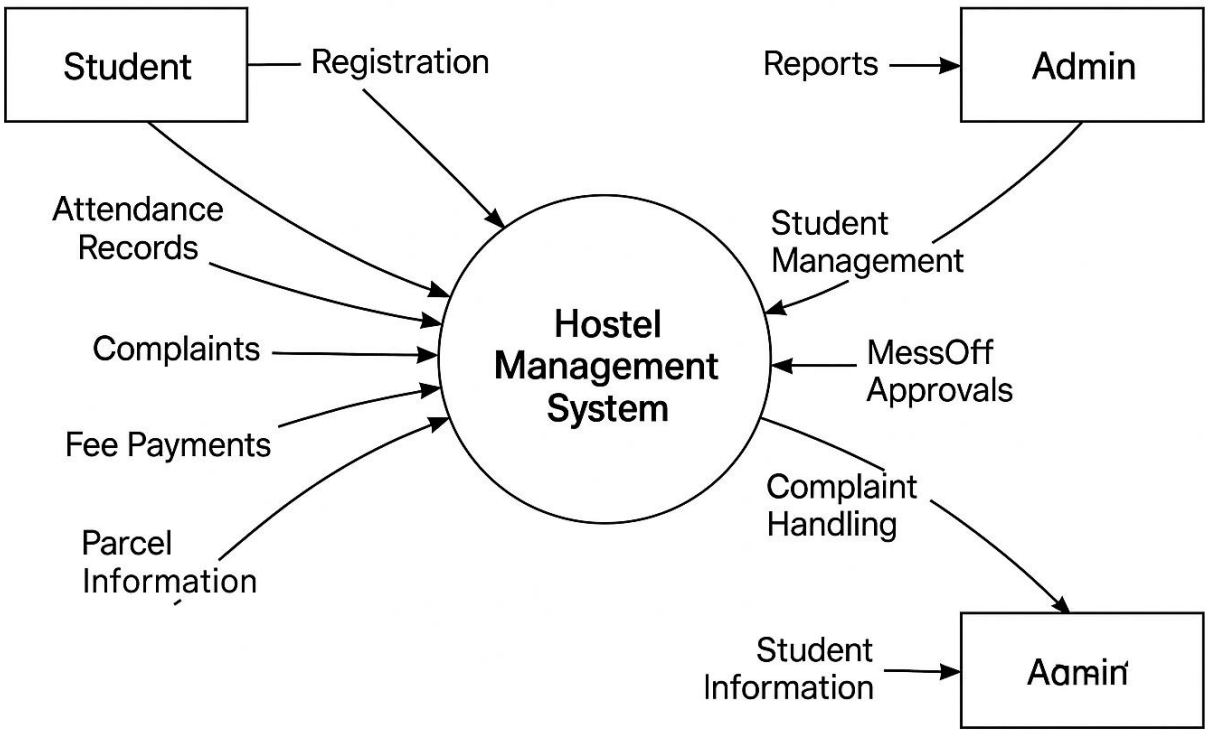
The Entity-Relationship Diagram (ERD) of the Hostel Management System defines the database structure by identifying entities such as Student, Admin, Complaint, Attendance, MessOff, Parcel, and Staff, along with their attributes and relationships. Each entity is connected through primary and foreign keys to ensure data consistency and avoid redundancy. For example, a student can submit multiple complaints, apply for mess-off, or receive parcels, while the admin manages and student records. The ERD provides a clear visual representation of these relationships, serving as a blueprint for database design and ensuring that hostel operations like registration, attendance tracking, fee management, and complaint handling are efficiently modeled.



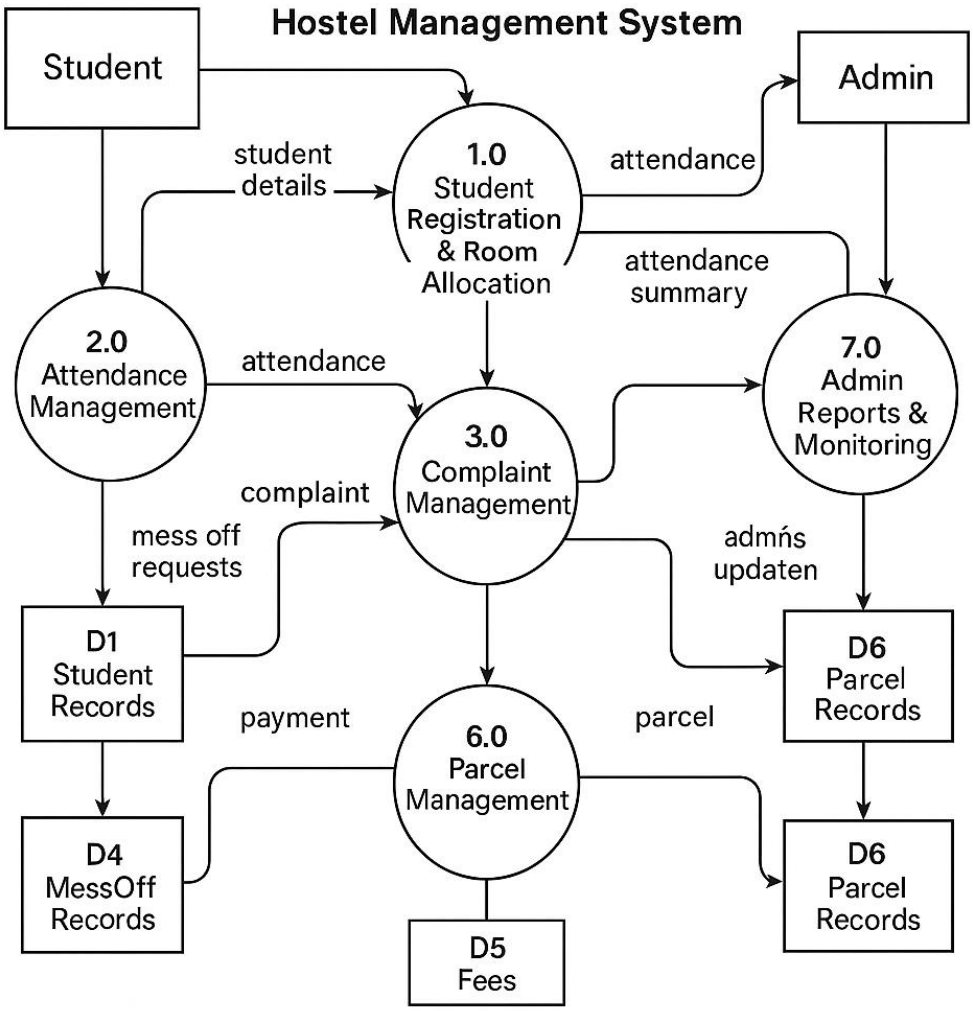
5.3 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is used to visually represent how data moves within the Hostel Management System. It illustrates the flow of information between different processes, entities, and data stores. The DFD helps to understand how students, admins, and staff interact with the system, how complaints, attendance, fee records, and mess-off requests are processed, and how information is stored and retrieved. By modeling these flows, the DFD provides clarity on the functional processes of the system, making it easier to identify data dependencies and improve efficiency**.**

DFD Level 0



DFD Level 1

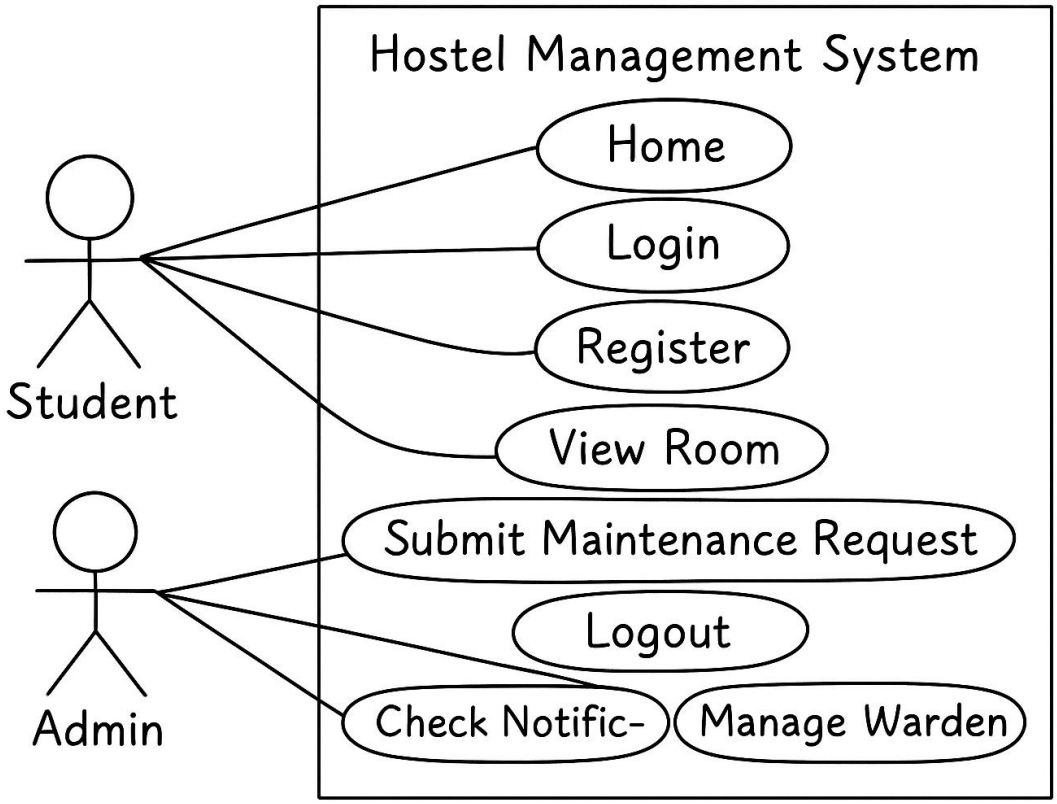


5.4 UML Diagrams (Use Case, Class, Sequence, Activity)

Unified Modeling Language (UML) Diagrams are used to visually represent the structural and behavioral aspects of the Hostel Management System. These diagrams help developers, students, and administrators understand how different system components interact with each other. In this project, four important UML diagrams are used: Use Case Diagram, Class Diagram, Sequence Diagram, and Activity Diagram. Each diagram provides a unique perspective, making the design more clear and easier to implement

1. Use Case Diagram

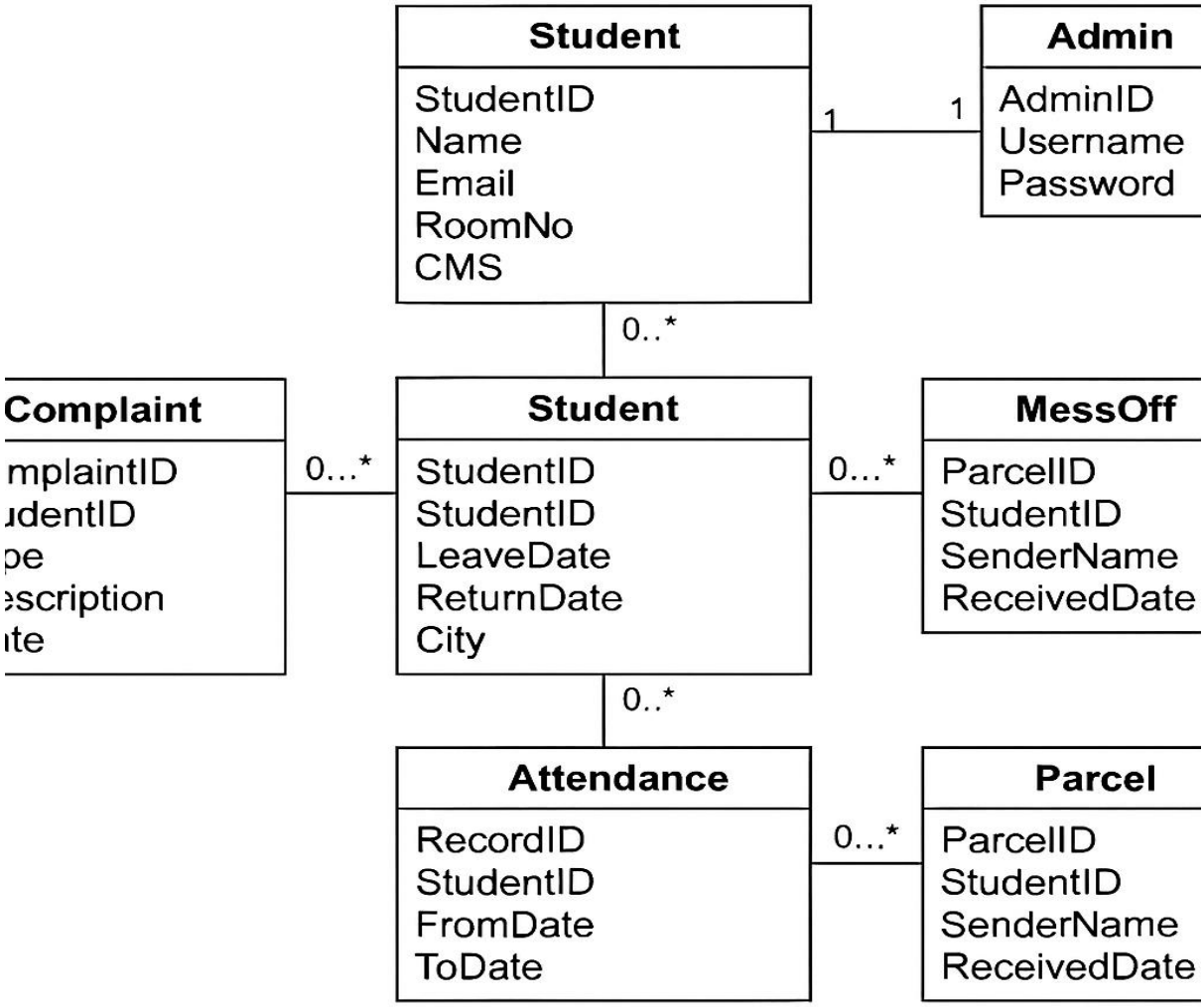
The Use Case Diagram illustrates the functional requirements of the Hostel Management System from the users' point of view. It shows how students, admins, interact with the system. For example, students can register, log in, apply for mess-off, and submit complaints; admins can manage students, staff salaries, complaints, and generate reports; while staff interact with salary management. The diagram captures these high-level interactions between users (actors) and system functionalities (use cases).



2. Class Diagram

The Class Diagram describes the static structure of the Hostel Management System by showing system classes, attributes, methods, and relationships. For example, the student class contains attributes such as StudentID, Name, Email, RoomNo, and CMS, while the Admin class

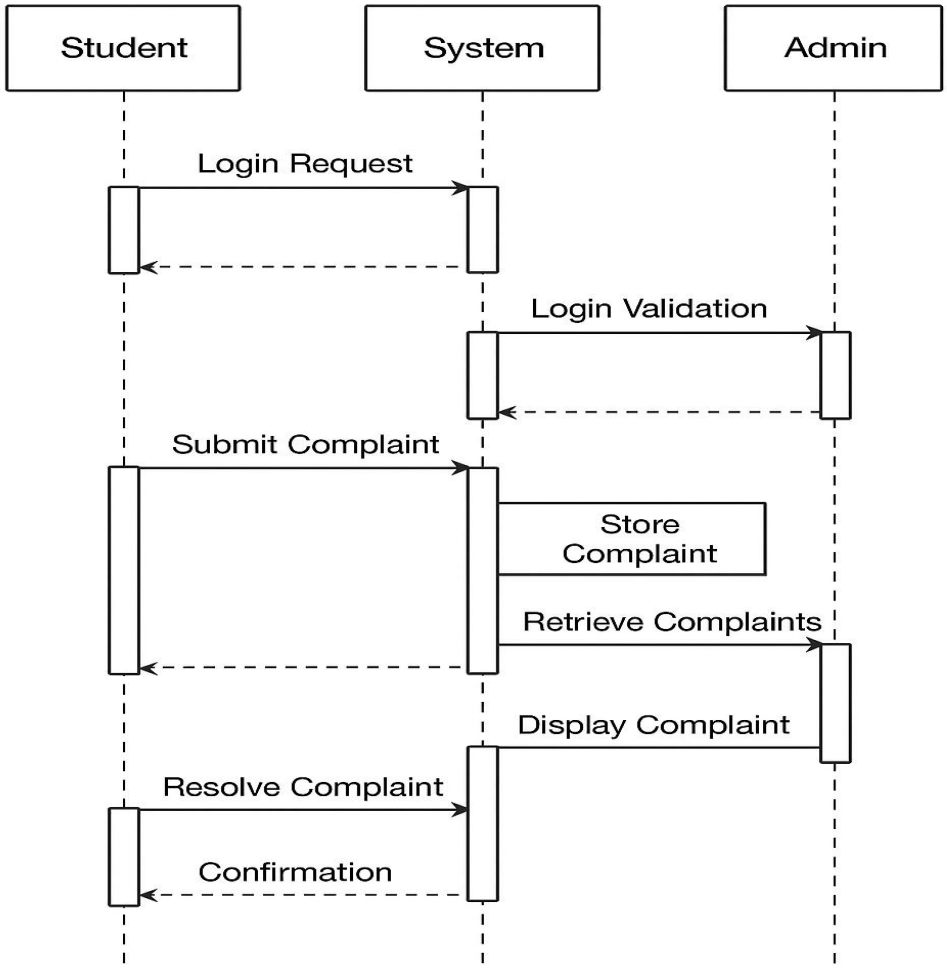
has AdminID, Username, and Password. Relationships like “Student submits Complaint”, and “Student applies MessOff” are shown with associations. This diagram helps in understanding how different objects in the system are linked and how data flows among them.



3. Sequence Diagram

The Sequence Diagram models the dynamic behavior of the Hostel Management System by showing the sequence of interactions between objects over time. It highlights how operations such as student login, complaint submission, and admin approval occur step-by-step. For example, a student logs in → system validates credentials → system displays dashboard →

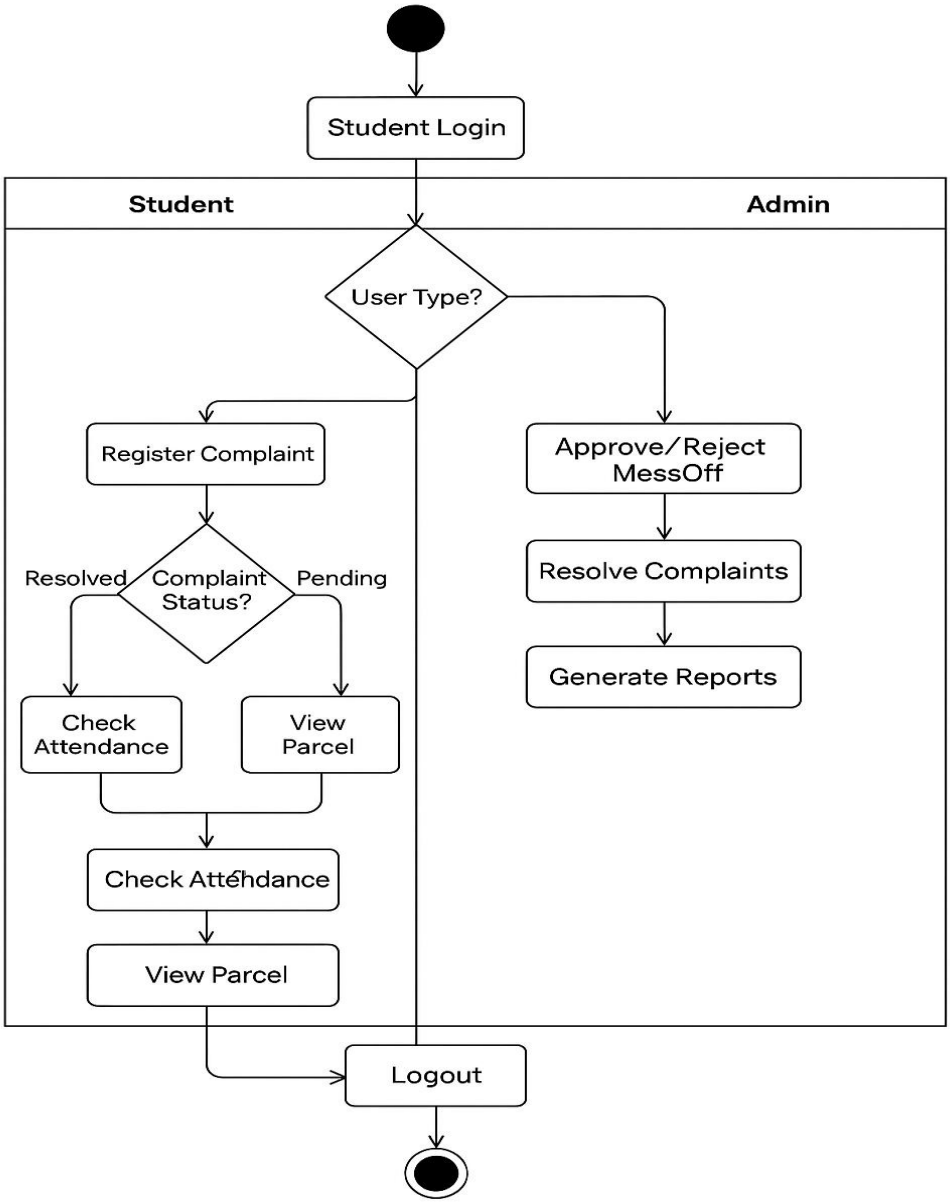
student submits complaint → system stores complaint → admin reviews complaint. The diagram is useful to understand the message flow and ordering of actions in the system.



4. Activity Diagram

The Activity Diagram describes the flow of activities performed in the Hostel Management System. It is a flowchart-style diagram that depicts how processes such as student registration, complaint submission, attendance management, and staff salary processing take place. For instance, the diagram may begin with a student login, followed by choices like apply

mess-off, submit complaint, or view parcel status. Concurrently, the admin can approve requests and generate reports. This diagram is particularly helpful in understanding workflows and decision points within the system.



Chapter 6

Implementation

6.1 Development Environment

The Hostel Management System was developed using a web-based environment with the following setup:

- **Frontend Technologies:** HTML5, CSS3, JavaScript, Bootstrap (for responsive design).
- **Backend Technology**: PHP (server-side scripting).
- **Database:** MySQL (for storing student, staff, and hostel records).
- **Server Environment:** XAMPP (Apache, PHP, and MySQL integration).
- **IDE/Editor Used:** Visual Studio Code.
- **Operating System:** Windows 10/11.
- **Browser for Testing:** Google Chrome, Mozilla Firefox.

This environment provided an easy-to-deploy, cost-effective, and scalable solution for development and testing.

6.2 Module Descriptions

The system is divided into multiple functional modules, each responsible for specific tasks:

1. **Student Module:**
    a. Handles student registration, login, and profile management.
    b. Allows students to apply for mess-off, submit complaints, and track attendance/parcel records.
2. **Admin Module:**

a. Manages student records, room allocations, and hostel activities.

b. Handles staff salaries, hostel expenses, and generates system reports.

c. Resolves student complaints and approves/rejects mess-off applications.

3. **Complaint Management Module:**

a. Allows students to register complaints

b. Admin views, updates, and resolves complaints.

4. **Attendance & Mess-Off Module:**

a. Tracks student leave requests and return dates.

b. Ensures proper hostel attendance monitoring.

5. **Parcel Management Module:**

a. Records parcels received for students.

Allows admin to notify students about parcel collection.

6.3 Database Design

The database is designed in MySQL to store and manage hostel-related data efficiently.

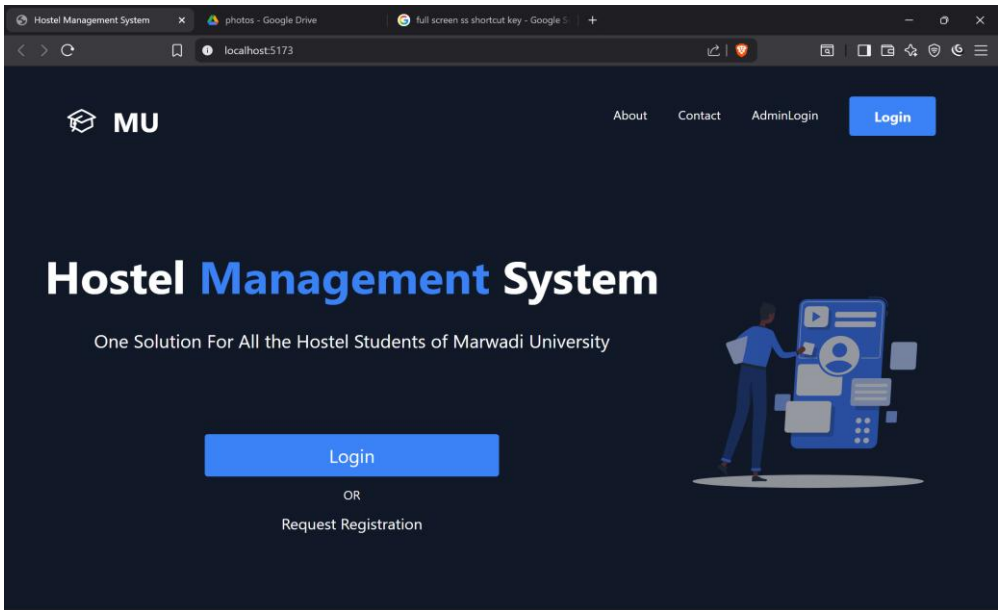
- **Tables Created:**
  - students (StudentID, Name, Email, RoomNo, Dept, CMS, Password)
  - admin (AdminID, Username, Password)
  - complaints (ComplaintID, StudentID, Type, Description, Date, Status)
  - attendance (RecordID, StudentID, LeaveDate, ReturnDate, City)

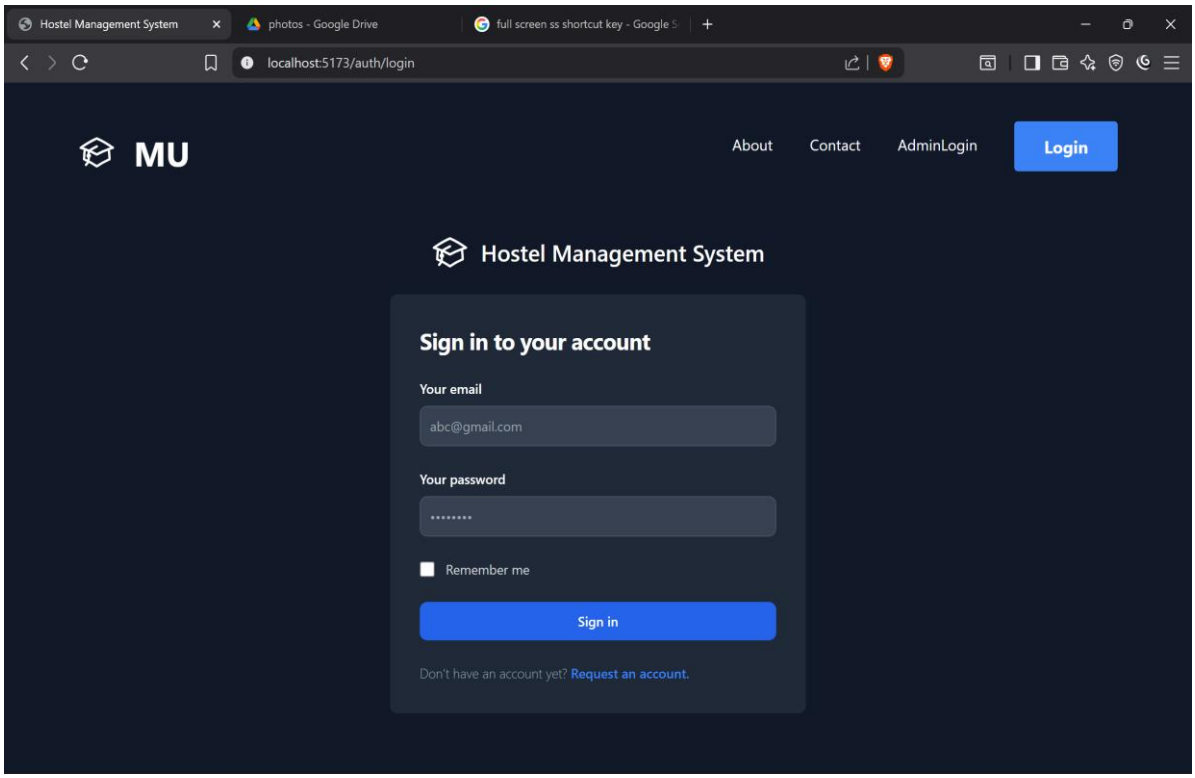
- mess_off (MessOffID, StudentID, FromDate, ToDate, Status)
- parcel (ParcelID, StudentID, SenderName, ReceivedDate)

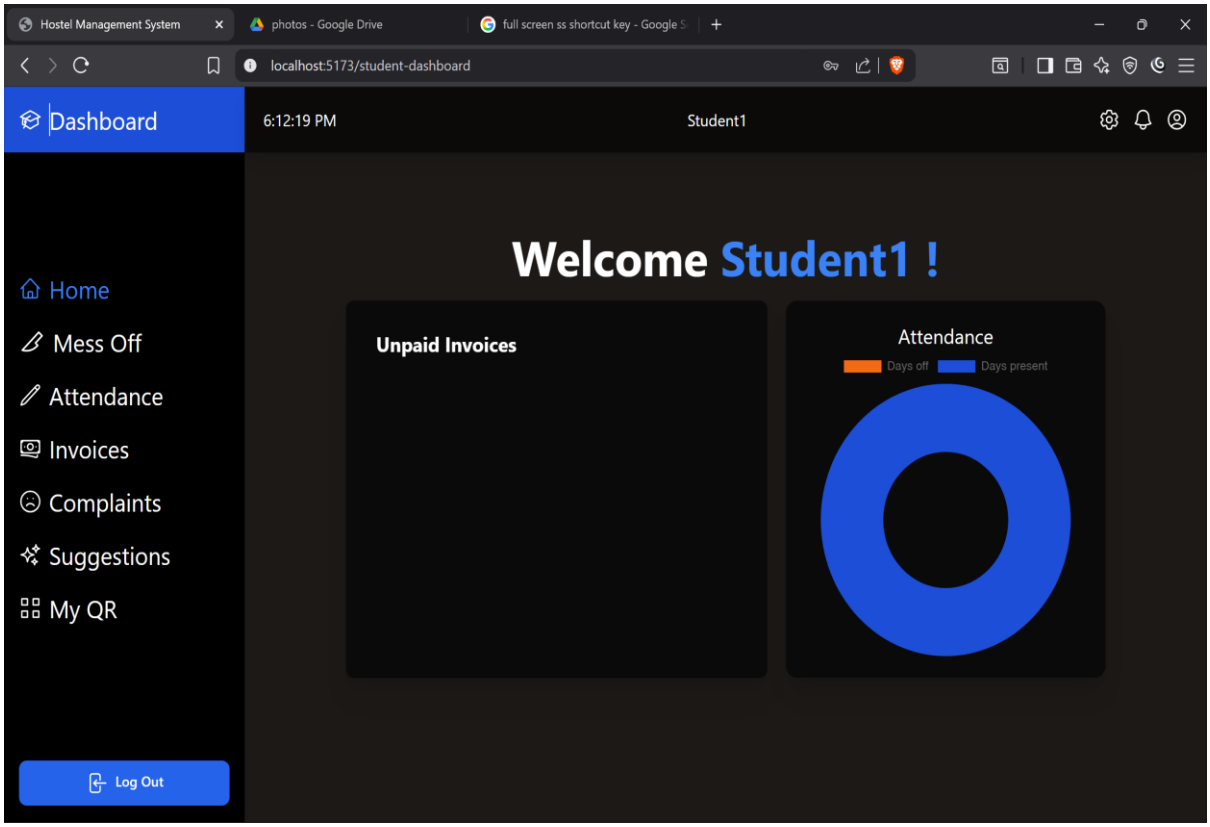
The database ensures referential integrity by establishing relationships using primary keys and foreign keys.

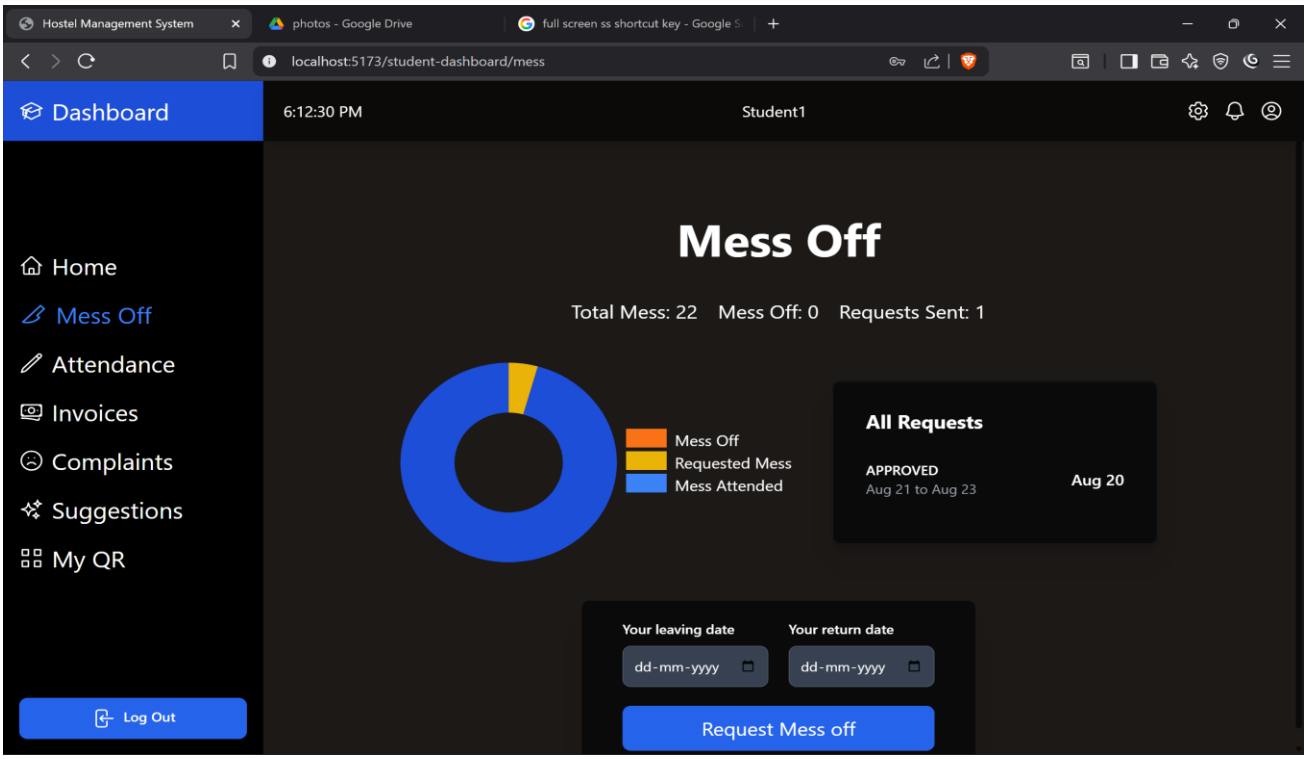
6.4 Screenshots of Implementation

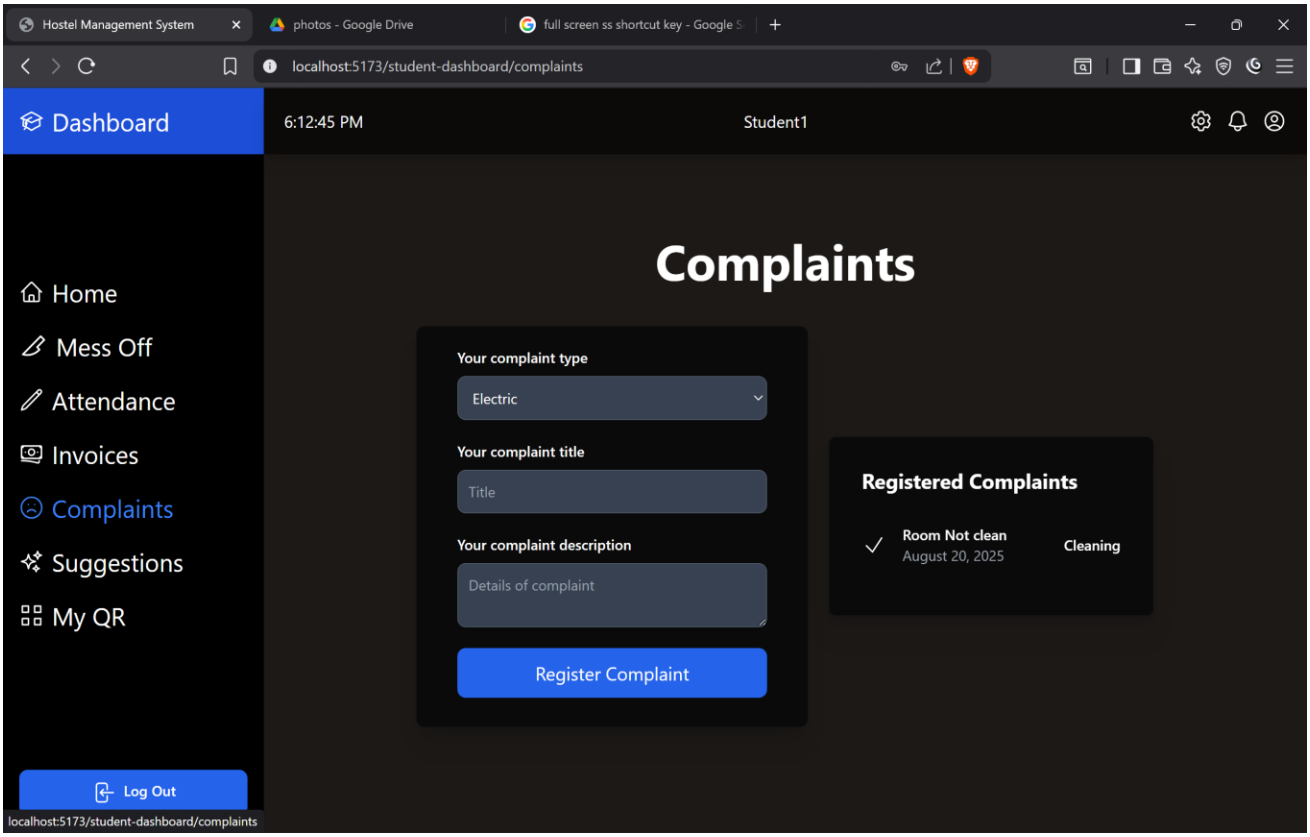
- **Login Page** – Student and Admin login interface.
- **Student Dashboard** – Overview of attendance, complaints, and parcels.
- **Admin Dashboard** – Controls for student management, and reports.
- **Complaint Form** – Student submits complaints with description.
- **Mess-Off Application Form** – Student applies for leave from hostel mess.
- **Attendance Records** – List of students' in-out records.
- **Parcel Management Page** – Details of parcels received for students.

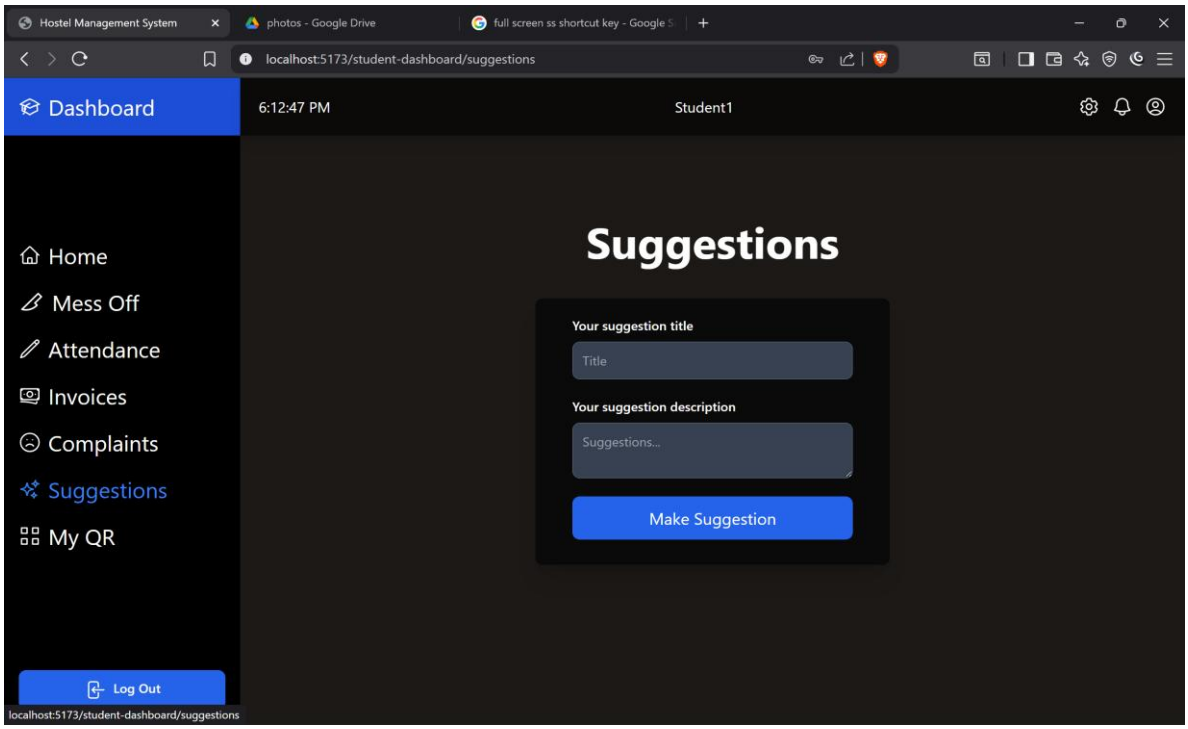


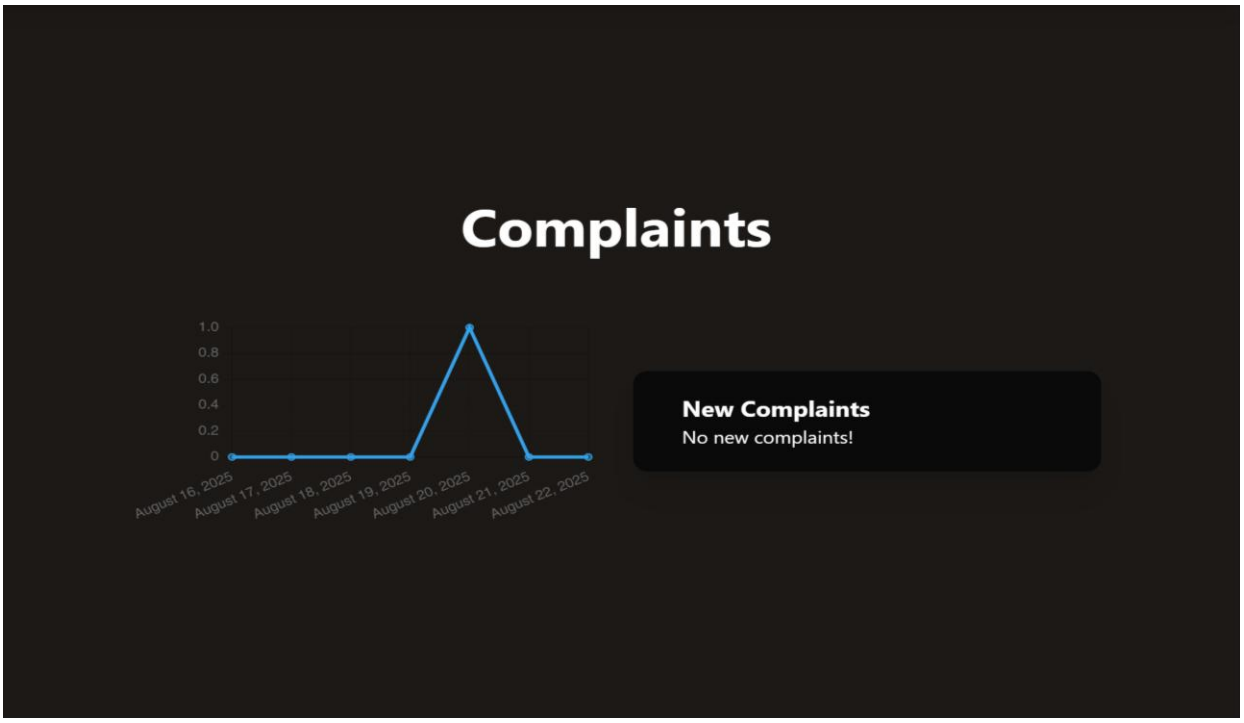

Hostel Management System
photos - Google Drive
full screen ss shortcut key - Google S
localhost:5173/auth/login
MU
About
Contact
AdminLogin
Login
Hostel Management System
Sign in to your account
Your email
abc@gmail.com
Your password
Remember me
Sign in
Don't have an account yet? Request an account.

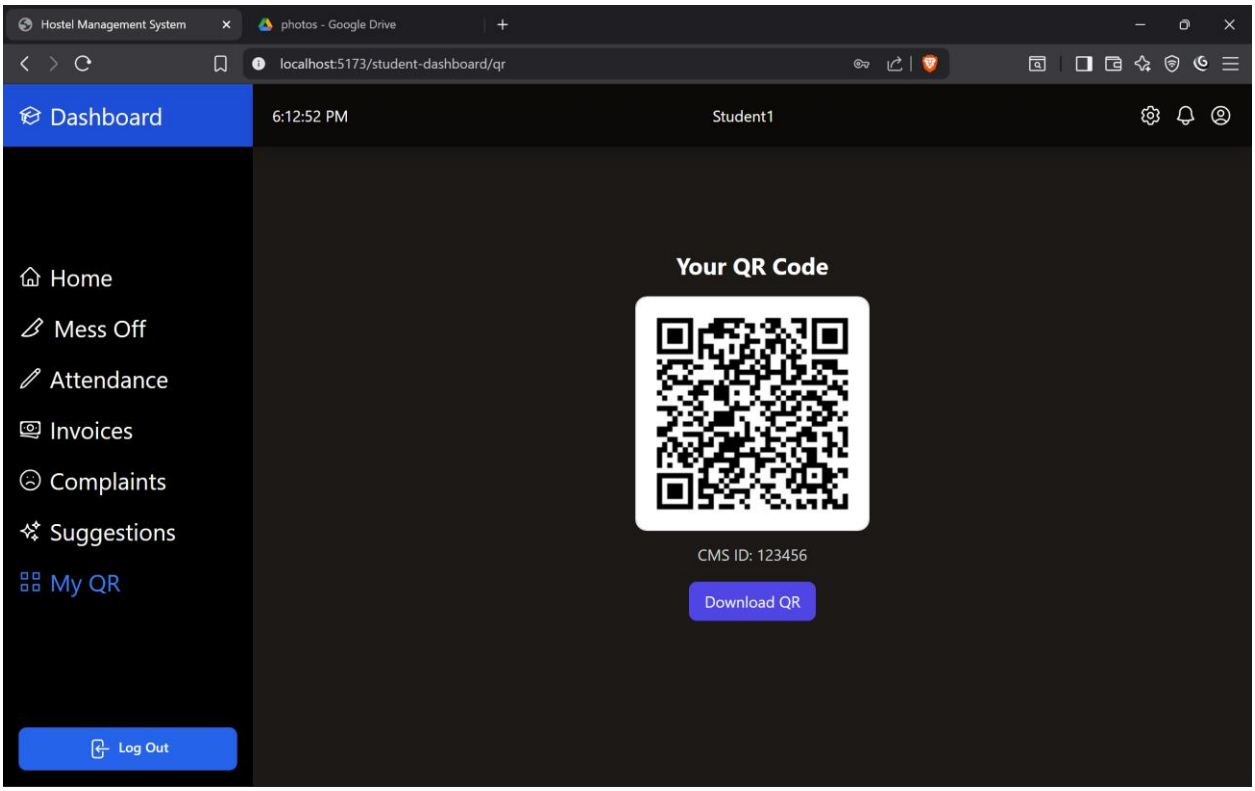

Hostel Management System
photos - Google Drive
full screen ss shortcut key - Google S
localhost:5173/student-dashboard
Dashboard
6:12:19 PM
Student1
Welcome Student1 !
Home
Mess Off
Attendance
Invoices
Complaints
Suggestions
My QR
Unpaid Invoices
Attendance
Days off
Days present
Log Out

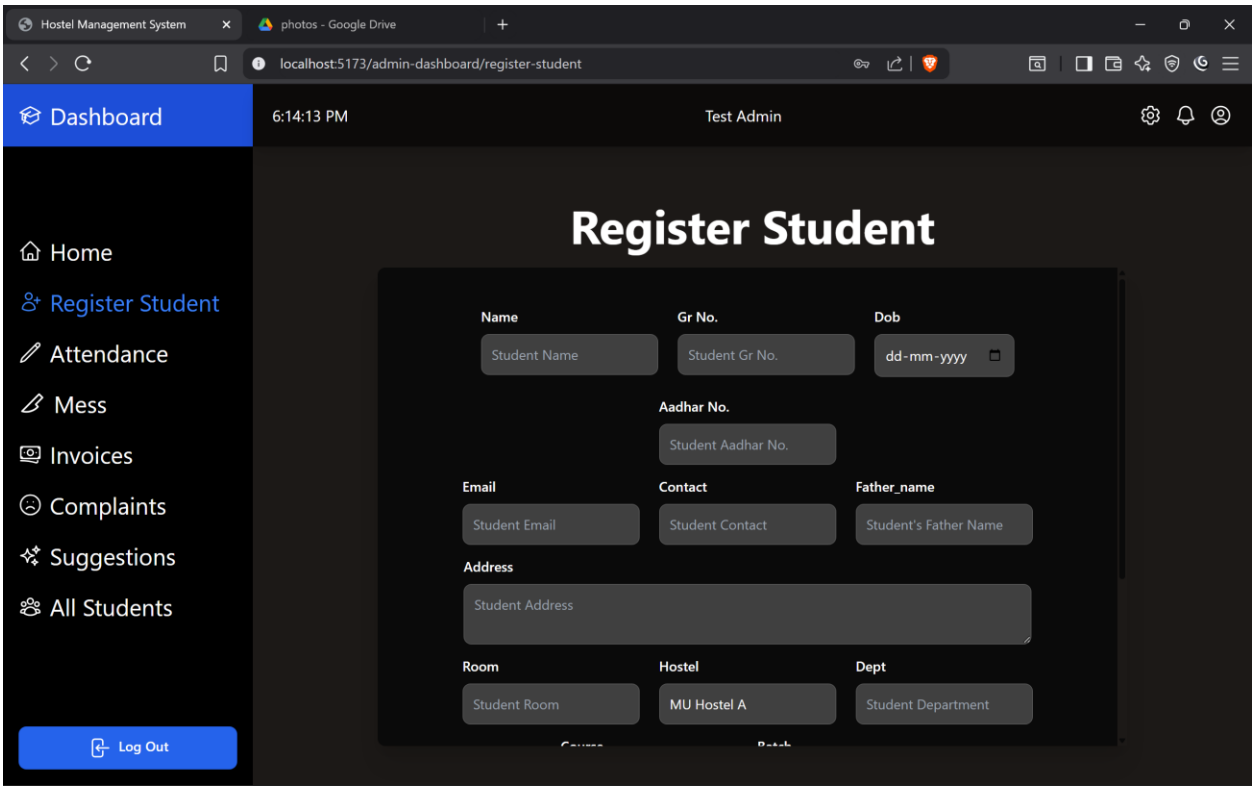

Hostel Management System
photos - Google Drive
full screen ss shortcut key - Google S
localhost:5173/student-dashboard/mess
Dashboard
6:12:30 PM
Student1
Home
Mess Off
Attendance
Invoices
Complaints
Suggestions
My QR
Log Out
Mess Off
Total Mess: 22
Mess Off: 0
Requests Sent: 1
Mess Off
Requested Mess
Mess Attended
All Requests
APPROVED
Aug 21 to Aug 23
Aug 20
Your leaving date
dd-mm-yyyy
Your return date
dd-mm-yyyy
Request Mess off

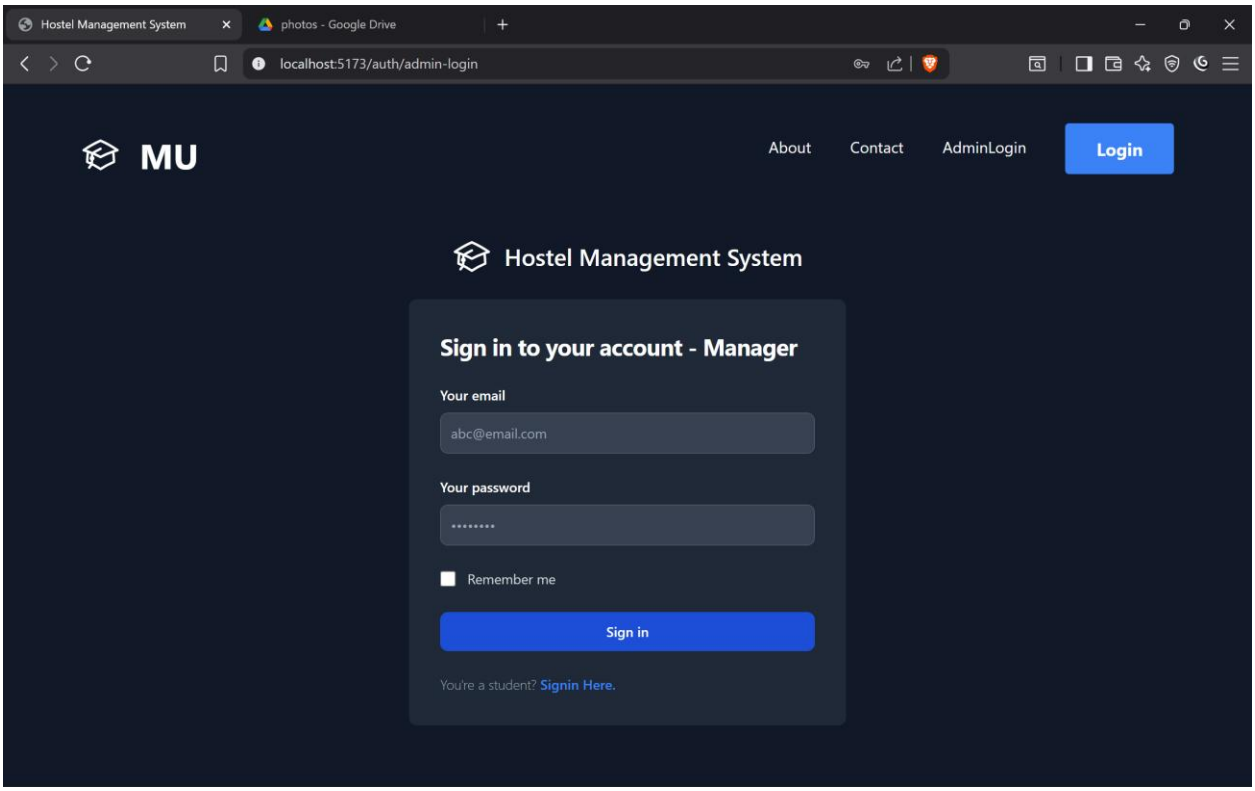

Hostel Management System
photos - Google Drive
full screen ss shortcut key - Google S
localhost:5173/student-dashboard/complaints
Dashboard
6:12:45 PM
Student1
Home
Mess Off
Attendance
Invoices
Complaints
Suggestions
My QR
Log Out
Complaints
Your complaint type
Electric
Your complaint title
Title
Your complaint description
Details of complaint
Register Complaint
Registered Complaints
Room Not clean
August 20, 2025
Cleaning
localhost:5173/student-dashboard/complaints

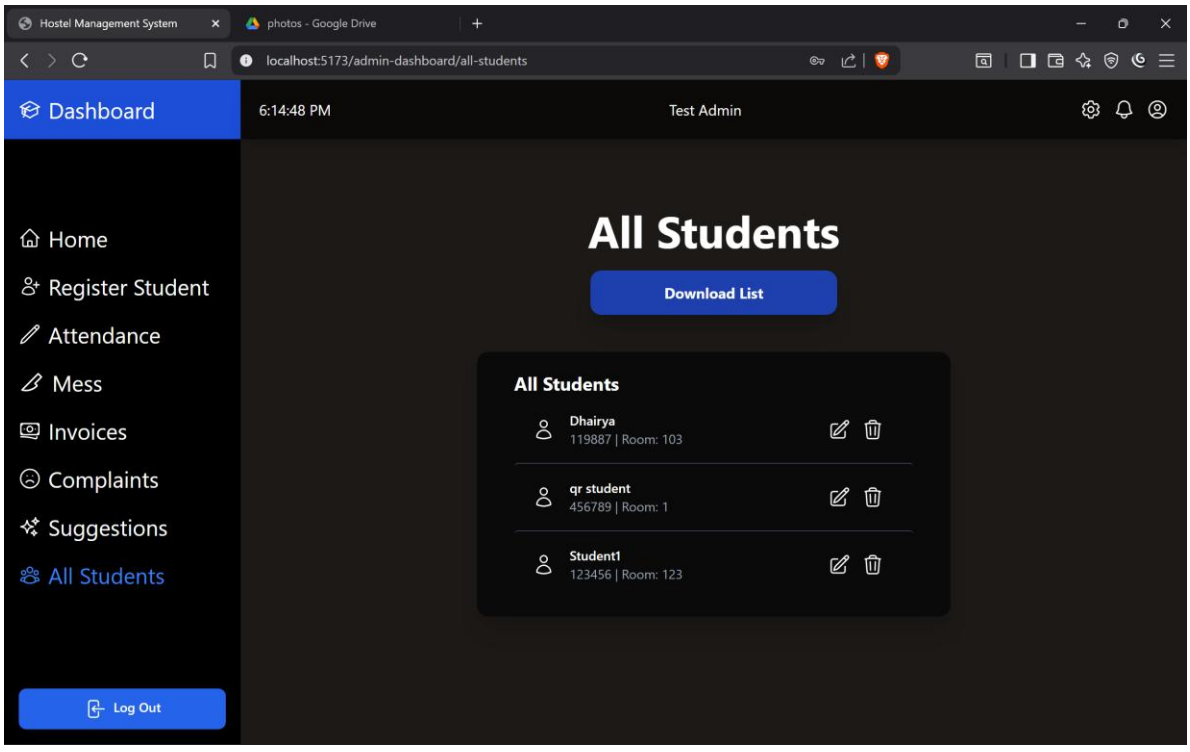

Hostel Management System
photos - Google Drive
full screen ss shortcut key - Google S
localhost:5173/student-dashboard/suggestions
Dashboard
6:12:47 PM
Student1
Home
Mess Off
Attendance
Invoices
Complaints
Suggestions
My QR
Log Out
localhost:5173/student-dashboard/suggestions
Suggestions
Your suggestion title
Title
Your suggestion description
Suggestions...
Make Suggestion

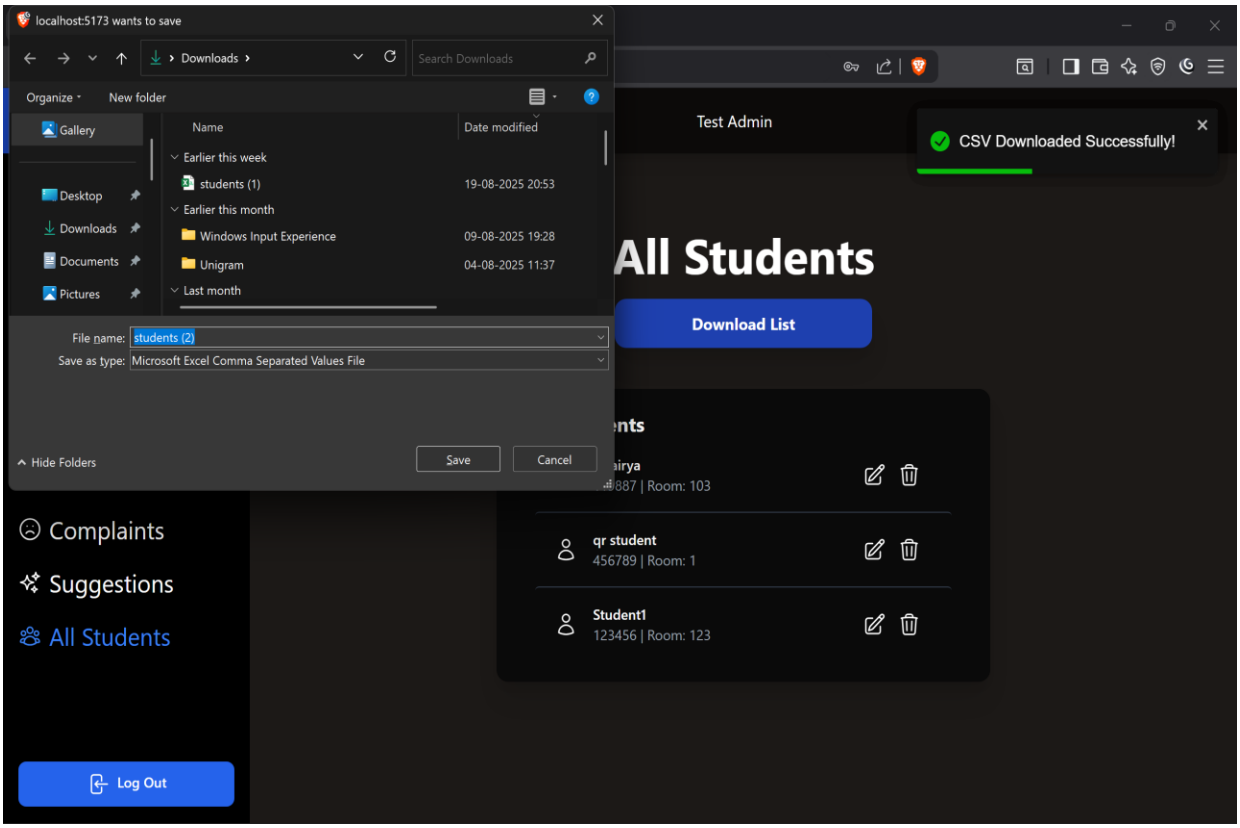

Complaints
1.0
0.8
0.6
0.4
0.2
0
August 16, 2025
August 17, 2025
August 18, 2025
August 19, 2025
August 20, 2025
August 21, 2025
August 22, 2025
New Complaints
No new complaints!

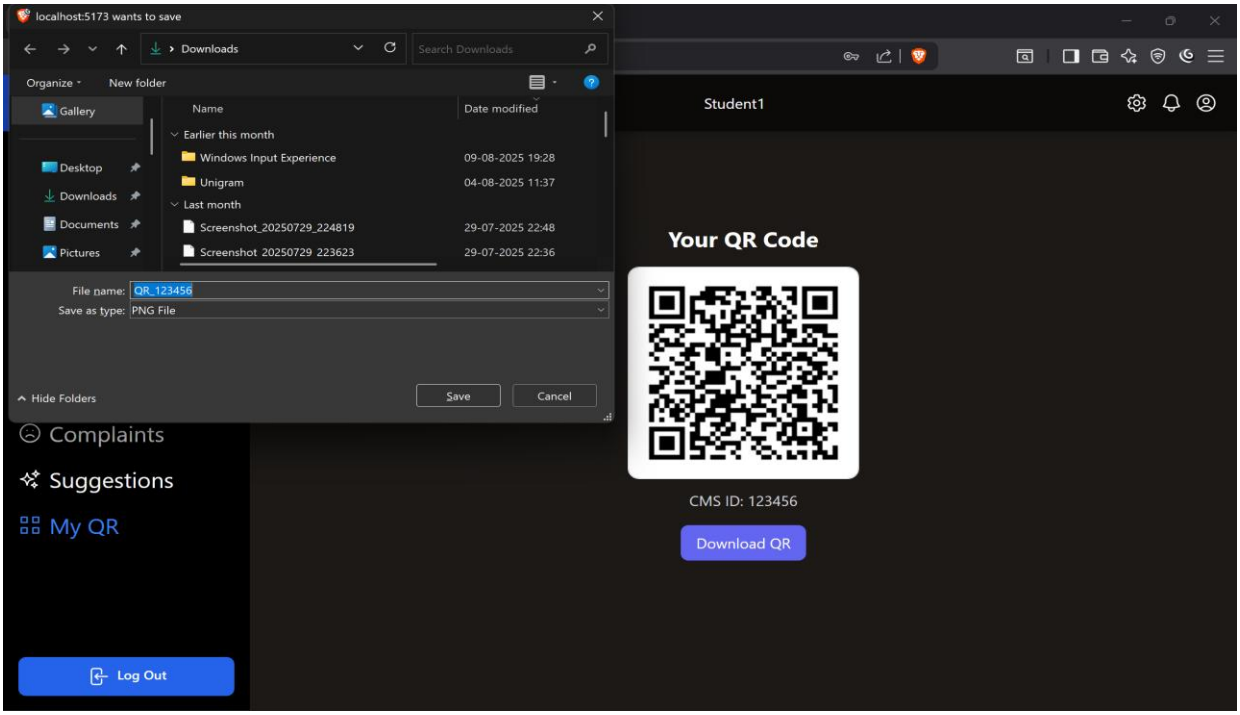

Hostel Management System
photos - Google Drive
localhost:5173/student-dashboard/qr
Dashboard
6:12:52 PM
Student1
Home
Mess Off
Attendance
Invoices
Complaints
Suggestions
My QR
Log Out
Your QR Code
CMS ID: 123456
Download QR


Hostel Management System
photos - Google Drive
localhost:5173/admin-dashboard/register-student
Dashboard
6:14:13 PM
Test Admin
Home
Register Student
Attendance
Mess
Invoices
Complaints
Suggestions
All Students
Log Out
Register Student
Name
Student Name
Gr No.
Student Gr No.
Dob
dd-mm-yyyy
Aadhar No.
Student Aadhar No.
Email
Student Email
Contact
Student Contact
Father_name
Student's Father Name
Address
Student Address
Room
Student Room
Hostel
MU Hostel A
Dept
Student Department


Hostel Management System
photos - Google Drive
localhost:5173/auth/admin-login
MU
About
Contact
AdminLogin
Login
Hostel Management System
Sign in to your account - Manager
Your email
abc@email.com
Your password
Remember me
Sign in
You're a student? Signin Here.


Hostel Management System
photos - Google Drive
localhost:5173/admin-dashboard/all-students
Dashboard
6:14:48 PM
Test Admin
Home
Register Student
Attendance
Mess
Invoices
Complaints
Suggestions
All Students
Log Out
All Students
Download List
All Students
Dhairya
119887 | Room: 103
qr student
456789 | Room: 1
Student1
123456 | Room: 123


localhost:5173 wants to save
Downloads
Search Downloads
Organize
New folder
Gallery
Desktop
Downloads
Documents
Pictures
Name
Date modified
Earlier this week
students (1)
19-08-2025 20:53
Earlier this month
Windows Input Experience
09-08-2025 19:28
Unigram
04-08-2025 11:37
Last month
File name: students (2)
Save as type: Microsoft Excel Comma Separated Values File
Hide Folders
Save
Cancel
Test Admin
CSV Downloaded Successfully!
All Students
Download List
qr student
456789 | Room: 1
Student1
123456 | Room: 123
Complaints
Suggestions
All Students
Log Out


localhost:5173 wants to save
Downloads
Search Downloads
Organize
New folder
Gallery
Desktop
Downloads
Documents
Pictures
Name
Date modified
Earlier this month
Windows Input Experience
09-08-2025 19:28
Unigram
04-08-2025 11:37
Last month
Screenshot_20250729_224819
29-07-2025 22:48
Screenshot 20250729 223623
29-07-2025 22:36
File name: QR_123456
Save as type: PNG File
Hide Folders
Save
Cancel
Student1
Your QR Code
CMS ID: 123456
Download QR
Complaints
Suggestions
My QR
Log Out

Chapter 7
Testing & Results

7.1 Testing Strategy

- To ensure the Hostel Management System performs reliably and meets user expectations, the following testing strategies were adopted:
- **Unit Testing**
    - Each module (e.g., Student Registration, Complaints, Attendance, Staff Salary) was tested individually.
    - Ensured that small components function correctly before integration.
- **Integration Testing**
    - Verified the interaction between different modules.
    - For example, when a student submits a complaint, it should correctly reflect in the admin dashboard.
- **System Testing**
    - The complete system was tested as a whole to check end-to-end workflows.
    - This included login → perform operation → database update → admin verification.
- **Functional Testing (Black Box Testing)**
    - Focused on inputs and outputs without checking the internal logic.
    - Example: Providing invalid login credentials should display an error message.
- **Performance Testing**
    - Checked how the system behaves under heavy load (e.g., multiple students logging in and submitting complaints simultaneously).
    - Verified database response times and page load efficiency.
- **Security Testing**
    - Ensured that only authenticated users (students and admins) can access the system.
    - Passwords were tested to confirm they are stored securely (hashed) and protected from SQL injection attacks.
- **User Acceptance Testing (UAT)**
    - Conducted with actual hostel staff and students.
    - Tested for usability, clarity of forms, and effectiveness of features.
    - Feedback helped refine UI and improve navigation.
- **Compatibility Testing**
    - Verified that the system works on multiple browsers (Chrome, Edge, Firefox) and devices (laptops, mobiles).

Ensured responsiveness of UI across screen sizes

7.2 Test Cases

Test ID	Test Condition	Expected Output	Actual Output	Remark
TC-1	User enters valid username and password	User successfully logs into the Hostel Management System (dashboard opens)	User successfully logged in	Working
TC-2	User enters invalid username and/or password	System displays error message: “Invalid credentials” and stays on login page	Error displayed	Working
TC-3	User (student) views Room Availability from the list / search	List of rooms with details (room no., type, occupancy, rent, status) is displayed	Room details displayed	Working
TC-4	Admin assigns a room to a student (select student → select room → Allocate)	Room is assigned; student record shows assigned room; room status updates to Occupied	Room assigned and status updated	Working
TC-5	Admin deletes a student record (or removes student from hostel)	Student record removed; allocated room becomes Available	Student record deleted; room freed	Working
TC-6	Student scans QR code at hostel gate	System verifies QR code with student database → if registered, allow entry; else deny access	Entry allowed/denied correctly	Working
TC-7	Admin logs in with admin role	Admin features visible: allocate rooms, manage fees, view reports, etc.	Admin dashboard displayed	Working
TC-8	Student logs in with student role	Student can only view room details, fees, complaints, leave requests (no admin rights)	Limited access provided	Working

7.3 Results and Observations

After implementing and testing the Hostel Management System, the following results and observations were recorded:

1. Successful Login and Authentication
    a. Both admin and student login modules worked correctly with secure authentication.
    b. Unauthorized access attempts were blocked.
2. Efficient Student Registration and Room Allocation
    a. The registration process was smooth, and student details were stored properly in the database.
    b. Room allocation was accurate with no duplication errors.
3. Complaint Management System Functionality
    a. Students were able to submit complaints easily.
    b. Complaints were immediately reflected in the admin dashboard for timely resolution.
4. Attendance and Leave Records
    a. The system accurately recorded student attendance, leave requests, and return dates.
    b. Data retrieval was fast, and records were consistent across modules.
5. Mess-Off Module Accuracy
    a. Students could apply for mess-off, and records were updated in real-time.
    b. The admin was able to generate reports on mess-off requests for better mess management.
6. Parcel Management Module
    a. Parcels received for students were recorded correctly.
    b. Students could check parcel records without admin intervention.
7. Staff Salary Management
    a. Admin could add and update staff salary records easily.
    b. Salary reports were generated successfully and stored in the database.
8. Database Integrity
    a. All modules were properly integrated with the database.
    b. No data loss or corruption was observed during multiple operations.
9. Performance Results
    a. System performance remained stable under moderate load (20–30 users simultaneously).
    b. Page loading time was less than 2 seconds in most cases.
10. User Feedback

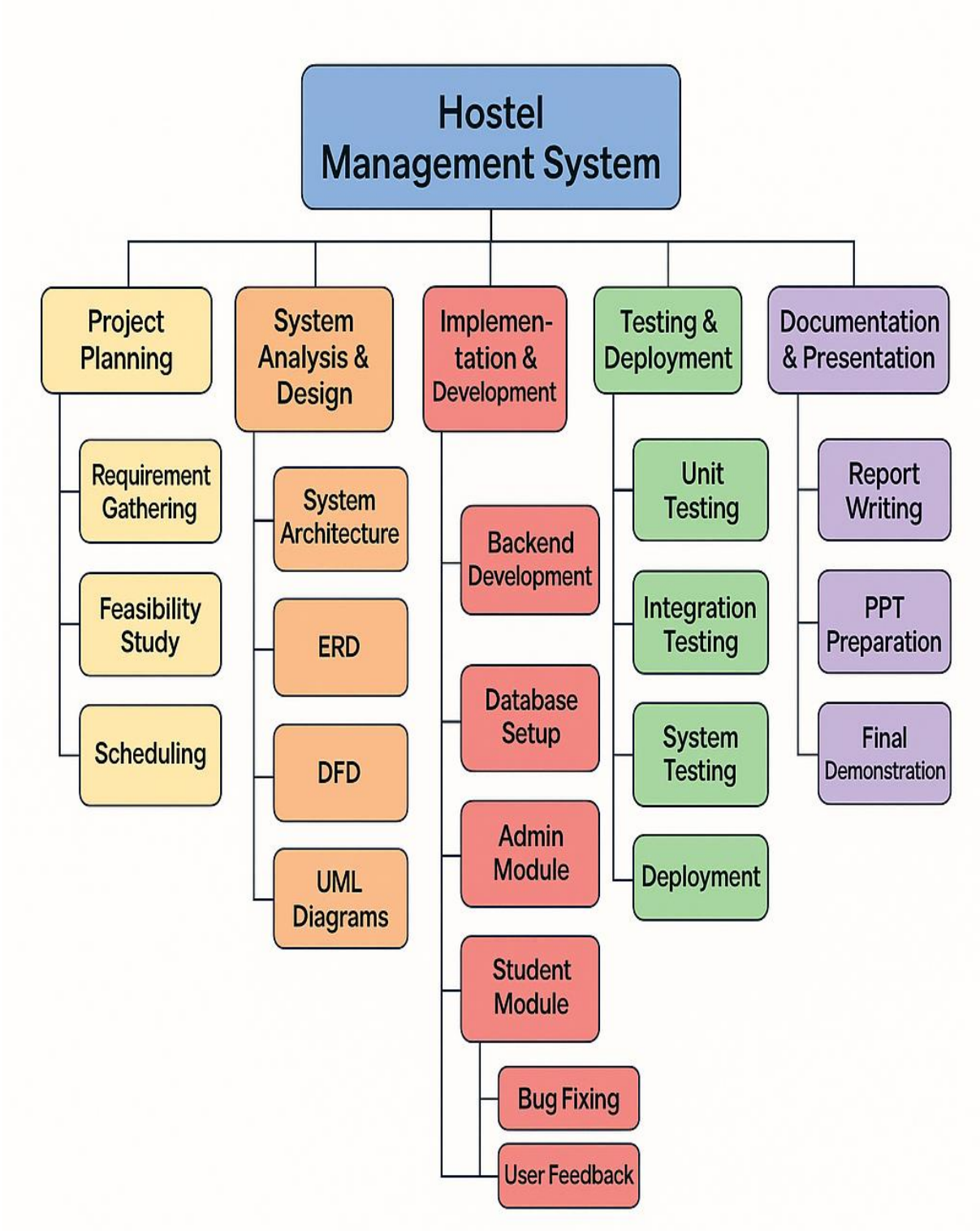
- . Suggestions included adding notification alerts (email/SMS) for complaints and leave approvals.

Chapter 8
Planning & Scheduling

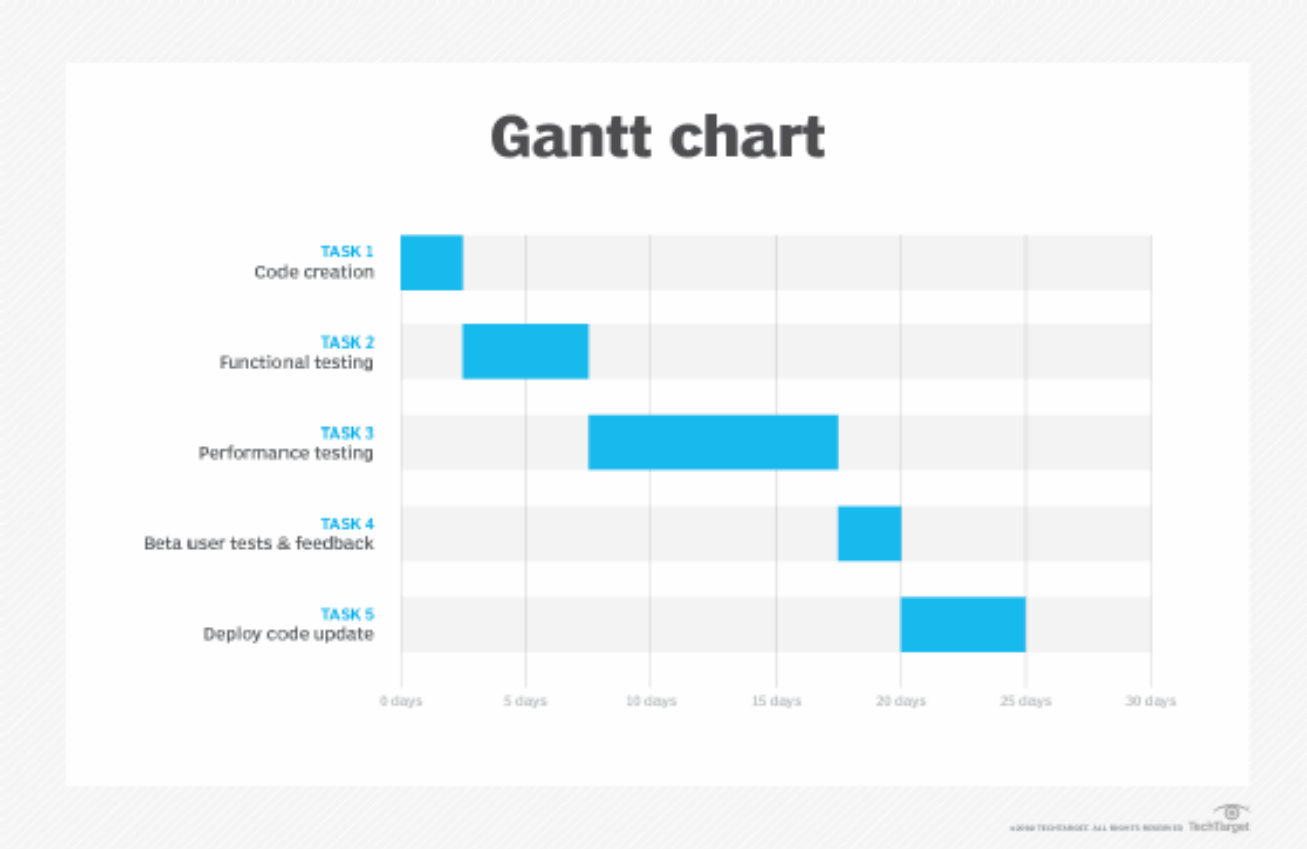
The Work Breakdown Structure (WBS) divides the Hostel Management System project into manageable tasks and phases. Each major task is further broken down into sub-tasks to ensure smooth project execution.

Work Breakdown Structure for Hostel Management System:

1. **Requirement Analysis**
    a. Collect requirements from stakeholders (students, admin, staff).
    b. Identify system modules (Student, Admin, Complaint, Attendance, Fee, etc.).
2. **System Design**
    a. Create system architecture.
    b. Prepare ERD, DFD, and UML diagrams.
3. **Database Design**
    a. Define tables (Student, Admin, Complaint, Attendance, Fee, Mess-Off, Parcel, Staff).
    b. Implement relationships and constraints.
4. **Frontend Development**
    a. Design user interface (login pages, dashboards).
    b. Create responsive forms for registration, complaints, and leave requests.
5. **Backend Development**
    a. Implement authentication & session management.
    b. Develop modules (student, admin, complaints, fee, salary, parcel).
6. **Integration & Testing**
    a. Integrate frontend with backend.
    b. Perform unit testing, integration testing, and user acceptance testing (UAT).
7. **Deployment**
    a. Deploy the system on localhost/server using XAMPP.
    b. Train users (admin, staff, students).
8. **Documentation & Report**
    a. Prepare technical documentation.
    b. Create project report and presentation.


Hostel Management System
Project Planning
System Analysis & Design
Implementation & Development
Testing & Deployment
Documentation & Presentation
Requirement Gathering
Feasibility Study
Scheduling
System Architecture
ERD
DFD
UML Diagrams
Backend Development
Database Setup
Admin Module
Student Module
Bug Fixing
User Feedback
Unit Testing
Integration Testing
System Testing
Deployment
Report Writing
PPT Preparation
Final Demonstration

8.2 Gantt Chart

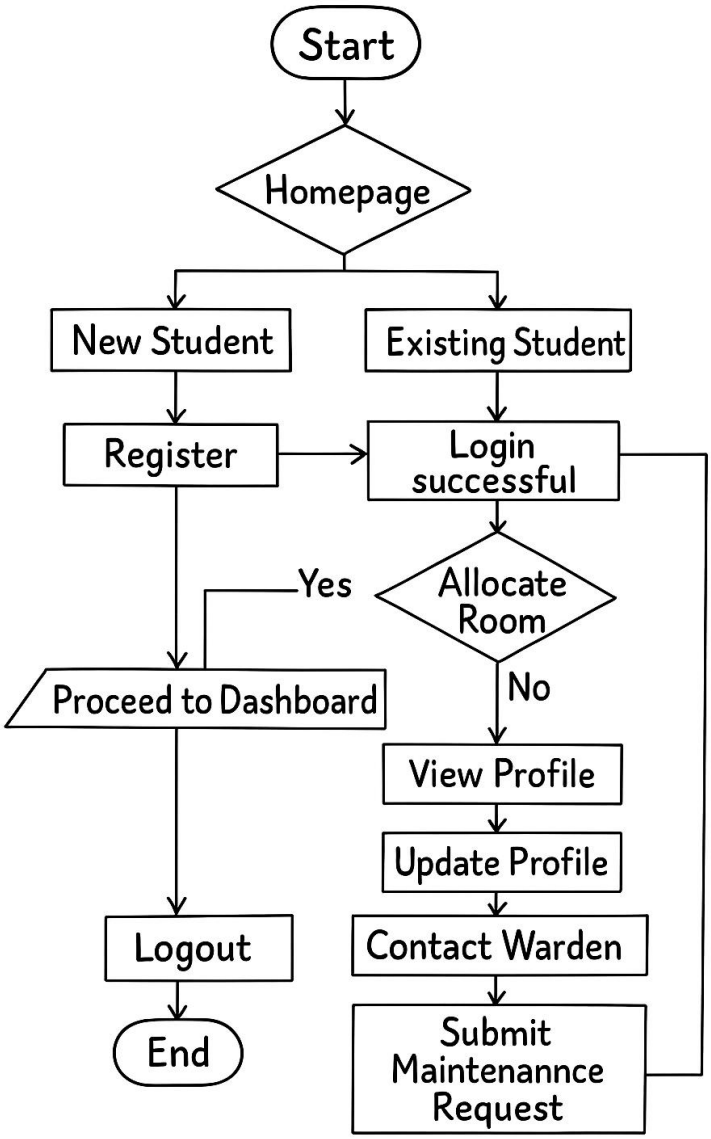


Chapter 9
Expected Outcomes

The Hostel Management System (HMS) is expected to deliver several tangible and intangible outcomes that will improve the efficiency, transparency, and overall management of hostel operations. By automating key processes such as student registration, fee management, complaint tracking, the system aims to minimize manual work, reduce errors, and save time for both administrators and students.

Expected Outcomes:

1. **Efficient Student Management** – Quick registration, room allocation, and centralized record keeping for all students.
2. **Automated Fee Tracking** – Real-time monitoring of hostel fee payments, pending dues, and receipt generation.
3. **Complaint Resolution System** – Faster redressal of student complaints with a systematic tracking mechanism.
4. **Mess and Attendance Management** – Digital handling of mess-off applications, attendance tracking, and visitor records.
5. **Enhanced Transparency** – Both students and admin have clear visibility of records, reducing disputes and errors.
6. **Report Generation** – Automatic generation of reports for hostel operations (e.g., student lists, fee reports, complaint logs).
7. **Time and Cost Saving** – Reduced paperwork, faster execution of routine tasks, and overall resource optimization.
8. **User-Friendly Interface** – Easy-to-use dashboard for administrators, students, and staff.
9. **Scalability** – The system can be extended in the future to handle multiple hostels or integrate with university ERP systems.


Start
Homepage
New Student
Existing Student
Register
Login successful
Yes
Allocate Room
Proceed to Dashboard
No
View Profile
Update Profile
Logout
Contact Warden
End
Submit Maintenannce Request

MANUAL vs. AUTOMATED HOSTEL MANAGEMENT SYSTEM

Manual System	Automated System
Paper-based student registration	Online student registration
Manual fee collection	Automated fee tracking
Complaints written in registers	Digital complaint management
Time-consuming mess/attendancecods	Online mess-off application
High chances of errors	Real-time transparency
High chances of errors	Real-time transparency

Chapter 10
Conclusion & Future Work

Conclusion

The Hostel Management System successfully addresses the limitations of traditional manual processes by introducing a centralized, automated solution. It streamlines hostel operations such as student registration, fee management, complaint handling, attendance tracking, and mess-off requests. The system ensures accuracy, reduces paperwork, improves efficiency, and provides transparency between students and administrators. Through a user-friendly interface and robust database design, the project demonstrates how technology can transform hostel administration into a faster, more reliable, and well-organized system.

Future Work

While the current system fulfills essential requirements, several enhancements can be incorporated in future versions:

- **Mobile Application Integration:** Developing Android/iOS apps for students and administrators to access services on-the-go.
- **Online Payment Gateway**: Enabling secure digital payments for hostel fees and other charges.
- **Notification System:** Adding SMS/Email alerts for fee deadlines, visitor approvals, or complaint updates.
- **AI-based Analytics:** Predicting occupancy rates, student attendance trends, and mess-off statistics using data analytics.
- **Cloud Deployment**: Hosting the system on cloud infrastructure for scalability, remote access, and real-time updates.
- **Biometric/Smart Card** Integration: Automating attendance and security through biometric authentication or RFID cards.

This project forms a strong foundation for modern hostel administration and can be expanded into a complete smart hostel management platform with further research and development.

References

1. MongoDB Documentation. (n.d.). Official MongoDB Documentation.Retrieved from https://www.mongodb.com/docs/ .
2. React.js Documentation. (n.d.).Retrieved from https://reactjs.org/ .
3. Node.js Documentation. (n.d.).Retrieved from https://nodejs.org/ .
4. Express.js Documentation. (n.d.).Retrieved from https://expressjs.com/ .
5. Marwadi Hostel Portal.

Consent for Filing Patent/Research Publication Application

We, Parth Shah , **Dhairya Aundhia, Rishi Patel, Kunj Nirmal** hereby give our full consent and authorization for the filing of a patent/research publication application for the project titled **"Hostel Management System**".

We hereby authorize Marwadi University and/or its legal representatives to file the patent/research publication application and act on our behalf regarding any matters related to this filing.

Date:
Name: Parth Shah
Signature:

Date: 30/08/2025
Name: Dhairya Aundhia
Signature:

Date: 30/08/2025
Name: Rishi Patel
Signature:

Date: 30/08/2025
Name: Kunj Nirmal
Signature: