



School of Computer Science, UPES, Dehradun

A Report File

On

Hostel and Mess Management System

BTECH – III Semester

Submitted By:

Name: Vaibhav Dobhal

SAP ID: 590017086

Batch : 30

Roll No. : 24010111776

Submitted To : Prof. Shareef Babu

HOSTEL AND MESS MANAGEMENT SYSTEM

Abstract

The Hostel and Mess Management System is a web-based administrative application designed to streamline hostel operations in educational institutions. Developed using HTML/CSS for the user interface, Python Flask for backend logic, and MySQL for database storage, the system centralizes essential functions such as student record management, room allocation, mess attendance tracking, fee monitoring, complaint handling, and search operations. The project integrates multiple modules accessible through an admin dashboard, each supported by structured database tables. This report provides an in-depth analysis of the system's objectives, architecture, implementation, webpage designs, limitations, and future scope based entirely on the uploaded project directory.

1. INTRODUCTION

Hostel management involves maintaining large volumes of information, including student details, room allocations, fee payments, attendance logs, and complaints. Manual systems are inefficient, error-prone, and lack transparency. This project provides a digital platform to simplify and centralize hostel and mess administrative tasks.

The system includes separate HTML pages for every operation, a backend to process user actions, and a database to store persistent records. Hosting all modules under one admin dashboard ensures usability and efficiency.

2. PROBLEM STATEMENT

Traditional hostel administration relies on paper registers or fragmented digital files. These systems suffer from:

- Difficulty in retrieving student and room details
- Errors during manual fee collection
- Missing mess attendance records
- No structured complaint mechanism
- No unified dashboard for administrators
- High dependency on physical registers

The proposed system digitizes these processes, reducing error rates and increasing accessibility.

3. OBJECTIVES

Based on the project files, the system aims to:

1. Create and maintain structured student records
2. Digitally allocate and track hostel room occupancy
3. Log mess attendance for meals on a daily basis
4. Record hostel and mess fee payments
5. Maintain a complaint submission system
6. Provide a centralized dashboard for navigation
7. Enable quick search of student data using roll number or name
8. Use a relational database for secure, long-term storage

4. SYSTEM ANALYSIS & ARCHITECTURE

4.1 Technology Stack

Frontend

- HTML5 for layout
- CSS3 (style.css) for design

Backend

- Python Flask framework for routing and request handling

Database

- MySQL relational database (hostel_mess.sql)

Architecture

A three-layer architecture:

User Interface (HTML/CSS) → Application Layer (Flask) → Database Layer (MySQL)

Each component communicates smoothly through HTTP requests and SQL queries.

4.2 File Structure Overview

Backend Folder

- A main script that defines routes for login, student management, room allocation, fee handling, mess attendance, complaints, and search.
- A database connector linking Flask to MySQL.

Frontend Folder

Contains individual webpages for:

- Login
- Dashboard
- Student operations
- Room operations
- Fee operations
- Mess attendance
- Complaints
- Search

A global CSS file ensures consistent design across all pages.

Database Folder

- A full SQL schema defining tables for students, rooms, fees, attendance, complaints, and admin credentials.

5. IMPLEMENTATION DETAILS (CONCEPTUAL EXPLANATION)

Here is how each module works internally.

Login System : The administrator enters credentials. The backend verifies them with stored admin data before granting access to the dashboard.

Student Management: Admin can add student details, which are stored in a database table. A separate page displays a full list of all students.

Room Allocation : Admin assigns rooms to students. These allocations are saved, and a page displays room occupancy records.

Fee Management : Admin records hostel/mess fee payments. These entries are stored and visible on the fee listing page.

Mess Attendance : Daily meal attendance (breakfast, lunch, dinner) is recorded and stored. Attendance logs can be viewed in table form.

Complaint Handling : Students can submit complaints via the admin form. All complaints are displayed in a listing page for administrative review.

Search System : Admin searches students by roll or name. Results are fetched from the student table and displayed.

6. DETAILED EXPLANATION OF EACH WEBPAGE

1. index.html — Admin Login Page

Purpose : This page is the system's entry point and ensures that only authorized users can access hostel operations.

Design

- Contains a visually centered login form
- Collects username and password
- Styled using the main CSS file

Flow : Admin enters credentials → Backend verifies → Access granted/denied.

Importance

Provides authentication and prevents unauthorized system use.

2.admin_dashboard.html — Admin Dashboard

Purpose : This is the central navigation hub for all operations.

Design

- Presents large, clearly labeled buttons linking to all modules
- Clean layout designed for quick administrative usage

Operations Accessible

- Add/View Students
- Allocate/View Rooms
- Add/View Fees
- Record/View Mess Attendance
- Add/View Complaints
- Search Students

Importance : Acts as the “home screen” for the entire platform.

3.add_student.html — Add Student Page

Purpose : Allows administrators to register a new student in the system.

Form Fields

- Student Name
- Roll Number
- Room Number
- Contact Number

Workflow : Admin fills student data → submits → backend stores it in the database.

Importance : Serves as the foundation for all other modules such as fees, attendance, and complaints.

4. view_students.html — Student List Page

Purpose : Displays all registered students.

Design

- A structured table listing ID, name, roll number, contact, and room
- Easy-to-read layout

Importance : Used frequently for verification, references, and management.

5. allocate_room.html — Room Allocation Page

Purpose : Assigns a room to a student.

Fields

- Roll Number
- Room Number

Workflow : Admin enters details → backend updates the room table.

Importance : Ensures that room records remain clear and organized.

6. view_rooms.html — View Room Allocation

Purpose : Displays a list of all rooms and the students assigned to them.

Design

A simple table showing:

- Room number
- Student roll assigned

Importance : Helps track occupancy and vacancy quickly.

7. fee.html — Add Fee Page

Purpose : Records fee payments for hostel and mess services.

Fields

- Roll Number
- Hostel Fee Amount
- Mess Fee Amount
- Payment Date

Flow : Admin enters info → backend adds the record to the fee table.

8. view_fees.html — Fee Records Page

Purpose : Shows all fee payment entries.

Design

Tabular listing of:

- Student roll
- Fee amounts
- Payment dates

Importance : Essential for financial verification and tracking.

9. mess_attendance.html — Record Mess Attendance

Purpose : Marks student presence/absence for daily meals.

Fields

- Roll Number
- Date
- Attendance Status

Use Case : Admin uses this page daily to log attendance.

10. view_mess_attendance.html — View Attendance

Purpose : Displays all attendance entries.

Design

A table containing:

- Roll Number
- Date
- Status

Importance : Useful for audits and mess fee calculations.

11. complaint.html — Register Complaint

Purpose : Allows admins to submit complaints collected from students.

Fields

- Student Name
- Roll Number
- Complaint Description

Importance : Centralizes grievance handling.

12.view_complaints.html — Complaint List

Purpose : Displays all complaints in a structured table.

Importance : Admin can track issues and respond efficiently.

13. search.html — Student Search Page

Purpose : Search for students by their name or roll number.

Design

- A search bar
- Results displayed below after backend processing

Importance : Enhances quick access to student info.

7. KEY FEATURES

The system supports:

- Full student record management
- Room allocation system
- Mess attendance monitoring
- Fee record management
- Complaint logging and viewing
- Search functionality
- Admin-only access
- Unified navigation dashboard

8. CHALLENGES & LIMITATIONS

- No update/delete options
- No validation for duplicates
- Basic security (passwords not encrypted)
- UI is static; no responsiveness
- No automatic reporting or analytics
- No student login functionality
- Room conflict checks missing

9. FUTURE SCOPE

Improvements could include:

- Editable and deletable records
- Role-based access (students, wardens)
- Better security with encrypted passwords
- Dynamic UI using modern frameworks
- Automated report generation
- Attendance graphs and fee charts

10. CONCLUSION

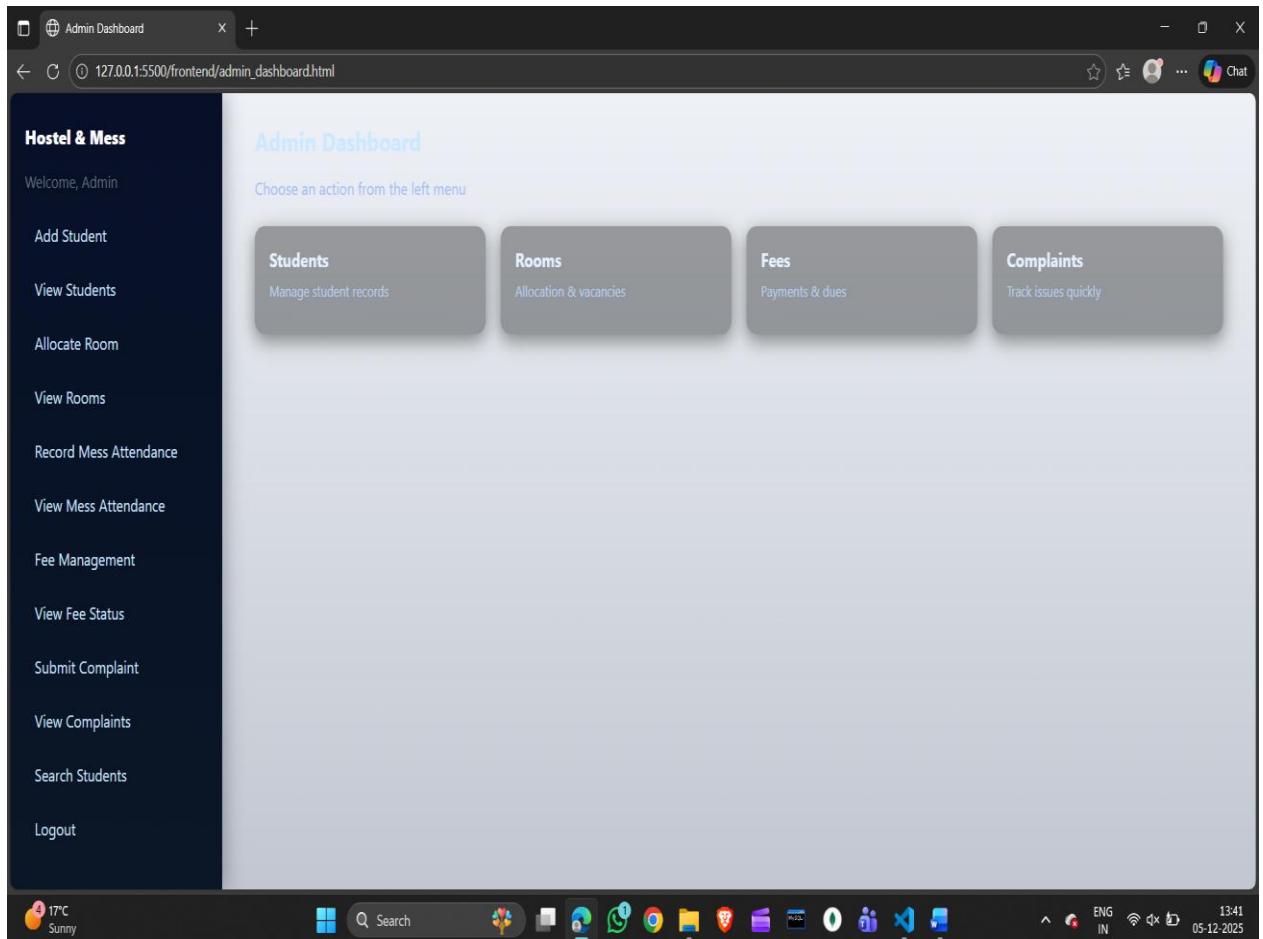
The Hostel and Mess Management System effectively demonstrates the principles and practical applications of Database Management Systems. Throughout the project, a complete relational database was designed, structured, and implemented using MySQL to manage multiple interconnected entities such as students, rooms, fees, mess attendance, complaints, and administrative credentials.

Each module in the system is supported by a well-defined database table, ensuring data consistency, referential integrity, and organized storage. The use of Flask as the backend allows for seamless interaction with the database through queries that insert, retrieve, and manage records, thereby highlighting real-world DBMS operations such as CRUD (Create, Read, Update, Delete), relational mapping, and transaction handling at an application level.

The project demonstrates DBMS concepts including schema design, primary key usage, data modeling, form-based data entry, backend-driven query execution, and structured retrieval through tabular views. By dividing functionality across multiple tables and linking them logically through shared attributes (such as roll numbers), the system showcases how relational databases efficiently manage interdependent data.

Overall, this project serves as a strong practical implementation of DBMS concepts, proving how database systems can be used to build reliable, scalable, and organized solutions for real-world institutional needs such as hostel and mess administration.

ADMIN DASHBOARD:



INFERENCE: The Admin Dashboard is the central control page of the Hostel & Mess Management System. It provides the administrator with quick access to all major modules such as student records, room allocation, fees, mess attendance, complaints, and search functions. The left sidebar offers direct navigation, while the main cards highlight key operational areas. This page helps the admin efficiently manage all hostel-related data through a unified interface.

ADD A STUDENT:

The screenshot shows a web browser window titled "Add Student" with the URL "127.0.0.1:5500/frontend/add_student.html". The page has a light gray background with a central white rectangular form. The form is titled "Add Student" and contains the instruction "Fill the form below to add a student". It includes five input fields: "Ashima Dobhal" (name), "77" (roll number), "8273011005" (contact), "ashima@gmail.com" (email), and "510" (room number). Below the form is a blue button labeled "Add Student" and a link "Back to Dashboard".

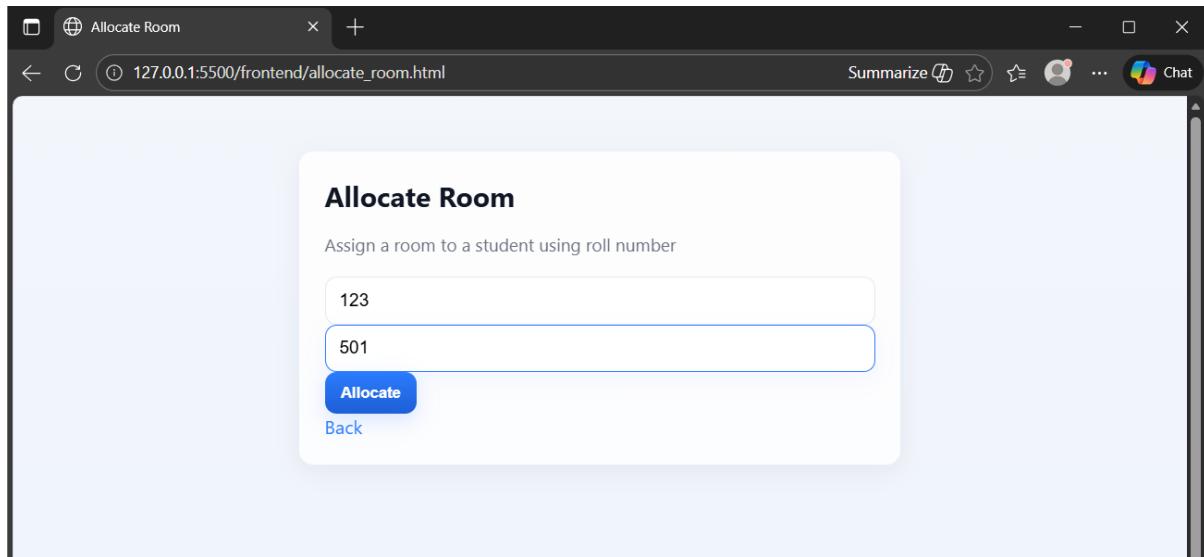
INFERENCE: The *Add Student* page allows the administrator to enter and submit new student details into the database through a form. By filling fields such as name, roll number, contact, email, and room number, the admin can register a new student in the system. This page represents the Create operation of the DBMS (CRUD).

VIEW STUDENT:

56	shareef babu	123456	104	1234567890	shareef@gmail.com
57	Ashima Dobhal	77	510	8273011005	ashima@gmail.com

INFERENCE: The *View Students* page displays all student records stored in the database in a tabular format. Each row shows a student's ID, name, roll number, room number, contact, and email. This page represents the Read operation of the DBMS (CRUD) and allows the admin to verify and track all student entries.

ALLOCATE ROOM:



The screenshot shows a web browser window titled "Allocate Room". The URL is 127.0.0.1:5500/frontend/allocate_room.html. The page has a central form with a title "Allocate Room" and a subtitle "Assign a room to a student using roll number". It contains two input fields: one with "123" and another with "501", both outlined in blue. Below the inputs is a blue "Allocate" button. At the bottom left of the form is a "Back" link.

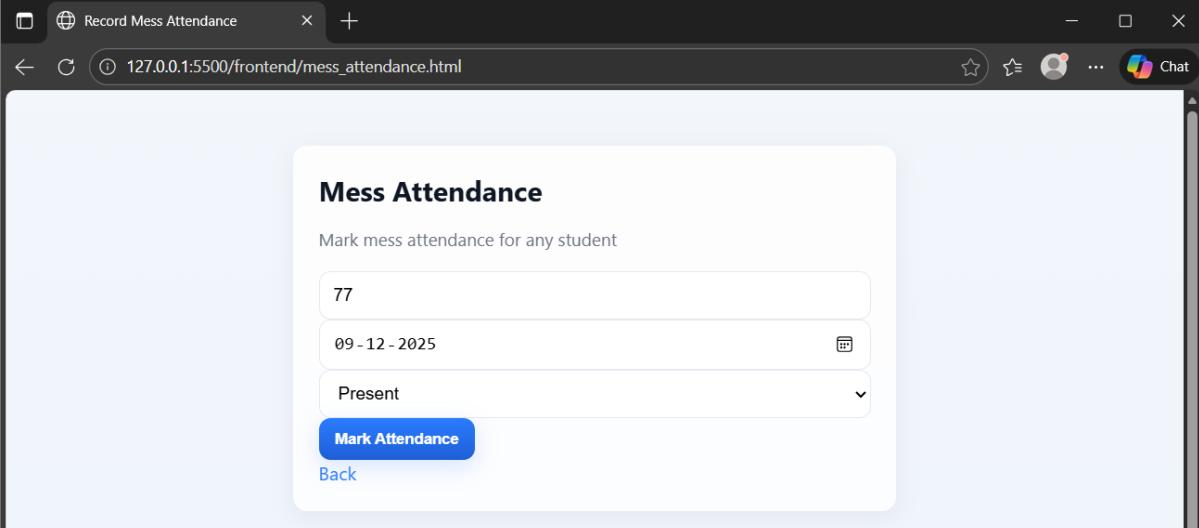
INFERENCE: The *Allocate Room* page allows the administrator to assign a room to a student by entering the student's roll number and the desired room number. Once submitted, the allocation is stored in the database. This page performs the Create/Update part of the DBMS workflow by linking students to room numbers.

VIEW ROOM:

	501	2	2	0	
	502	2	1	1	

INFERENCE: The *View Rooms* page displays all room allocations in a table format, showing how many students are in each room. This helps the administrator monitor occupancy and available rooms. It represents the Read operation of the DBMS by retrieving allocation records from the database.

RECORD MESS ATTENDANCE:



The screenshot shows a web browser window titled "Record Mess Attendance". The URL in the address bar is "127.0.0.1:5500/frontend/mess_attendance.html". The main content area has a title "Mess Attendance" and a subtitle "Mark mess attendance for any student". It contains three input fields: a text input with "77", a date input with "09 - 12 - 2025" and a calendar icon, and a dropdown menu set to "Present". Below these is a blue "Mark Attendance" button and a "Back" link.

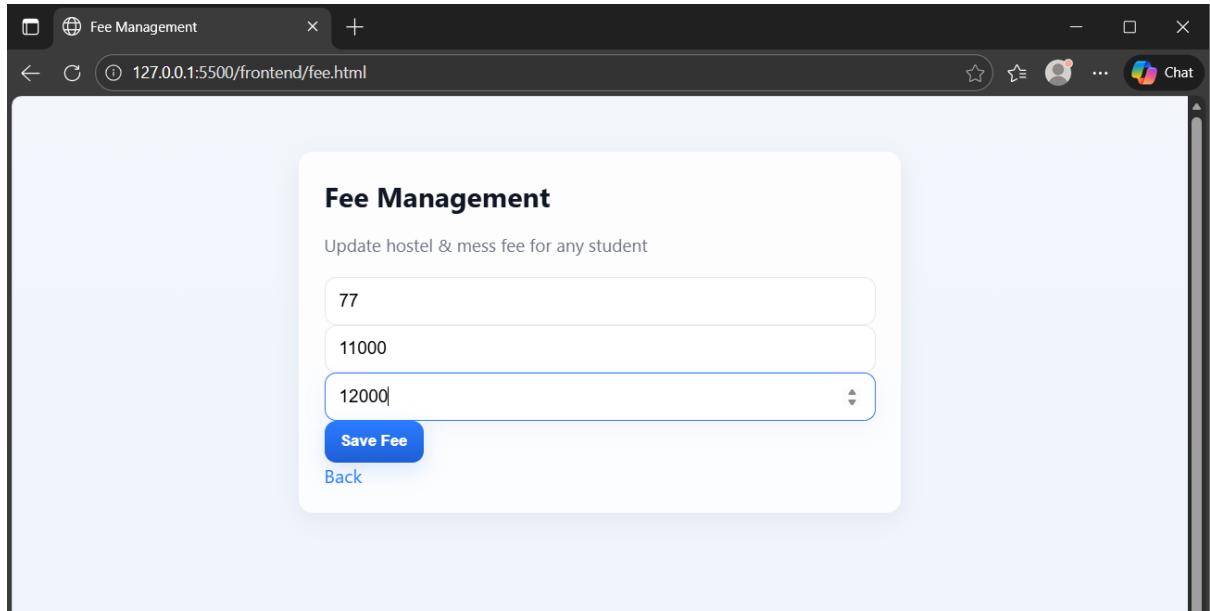
INFERENCE: The *Record Mess Attendance* page allows the administrator to mark whether a student is present or absent in the mess on a particular date. By entering the roll number, date, and attendance status, the system stores the attendance record in the database. This page represents the data entry (Create) operation for the mess attendance table.

VIEW MESS ATTENDANCE:

Wed, 03 Dec 2025 00:00:00 GMT	Mr. Ayush Kumar	123	Absent
Tue, 09 Dec 2025 00:00:00 GMT	Ashima Dobhal	77	Present

INFERENCE: The *View Mess Attendance* page displays all recorded mess attendance entries in a table format, including date, student name, roll number, and attendance status. It enables the administrator to review past attendance records quickly and reflects the Read operation in the DBMS workflow.

FEE MANAGEMENT:



The screenshot shows a web browser window titled "Fee Management" with the URL "127.0.0.1:5500/frontend/fee.html". The page has a light blue header with the title and a back button. Below the header is a white form area with a title "Fee Management" and a subtitle "Update hostel & mess fee for any student". The form contains three input fields: a dropdown menu with options "77", "11000", and "12000" (the last one is selected), a "Save Fee" button, and a "Back" link.

INFERENCE: The *Fee Management* page allows the administrator to enter or update a student's hostel and mess fee amounts by providing the roll number and corresponding fee details. Once submitted, the fee information is stored in the database. This page performs the Create/Update operation on the fee records.

VIEW FEE STATUS:

Aarav Sharma	BT23CSE001	22974	0	22974
Ashima Dobhal	77	23000	0	23000

INFERENCE: The *Fee Management* page allows the administrator to enter or update a student's hostel and mess fee amounts by providing the roll number and corresponding fee details. Once submitted, the fee information is stored in the database. This page performs the Create/Update operation on the fee records.

SUBMIT COMPLAINT:

The screenshot shows a web browser window titled "Submit Complaint" with the URL "127.0.0.1:5500/frontend/complaint.html". The main content is a "Submit Complaint" form. It has a text input field containing "77" and a larger text area containing "water leakage in washroom". Below these fields are two buttons: a blue "Submit" button and a blue "Back" button.

Roll Number	Student Name	Roll Number	Complaint Description	Status
28	Aarav Sharma	BT23CSE001	bad smell	Pending
29	Aarav Sharma	BT23CSE001	room dirty	Pending
30	Ashima Dobhal	77	water leakage in washroom	Pending

INFERENCE: The *Submit Complaint* page allows the administrator to record a student's hostel or mess-related complaint by entering the roll number and describing the issue. Once submitted, the complaint is stored in the database for further action. This page represents the Create operation in the DBMS complaint management module.

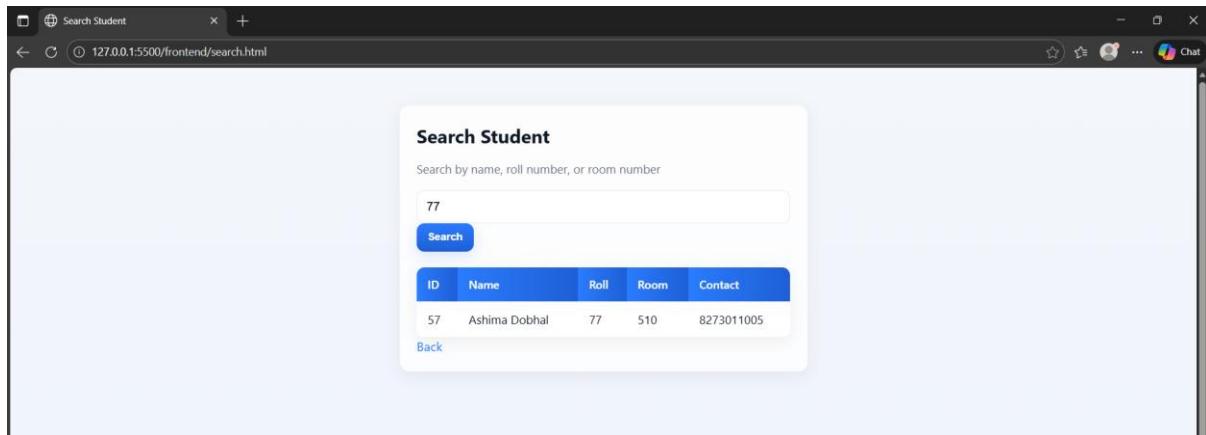
VIEW COMPLAINTS:

The screenshot shows a table of complaints recorded in the system. The table has columns for Roll Number, Student Name, Roll Number, Complaint Description, and Status. There are three entries:

Roll Number	Student Name	Roll Number	Complaint Description	Status
28	Aarav Sharma	BT23CSE001	bad smell	Pending
29	Aarav Sharma	BT23CSE001	room dirty	Pending
30	Ashima Dobhal	77	water leakage in washroom	Pending

INFERENCE: The *View Complaints* page displays all recorded complaints in a table format, showing details such as student name, roll number, complaint description, and status. It enables the administrator to monitor issues that require attention. This page reflects the Read operation of the DBMS and helps track unresolved complaints.

SEARCH STUDENTS:



INFERENCE : The *Search Students* page allows the administrator to quickly locate a student's details by entering a keyword such as roll number, name, or room number. The system retrieves matching records from the database and displays them in a table. This page demonstrates efficient data retrieval (Read/Search operation) in the DBMS, enabling fast access to specific student information.