A

Project Report

On

Hostel Booking System

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Shivam Kaushik	**2261526**
Rohit Joshi	**2261487**
Navneet Pandey	**2261384**
Abhinav Singh Bohra	**2261063**

Under the Guidance of

Mr. Anubhav Bewerwal

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI,

DISTRICT- NAINITAL-263132

2024-2025

STUDENT’S DECLARATION

We, **Shivam Kaushik, Rohit Joshi, Navneet Pandey** and **Abhinav Singh Bohra** hereby declare the work, which is being presented in the project, entitled **Hostel Booking System** in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (B.Tech.)** in the session **2024-2025**, is an authentic record of my work carried out under the supervision of **Mr. Anubhav Bewerwal**.

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date:

Shivam Kaushik
Rohit Joshi
Navneet Pandey
Abhinav Singh Bohra

CERTIFICATE

The project report entitled “Hostel Booking System” being submitted by Shivam Kaushik S/o Mr. Avendra Kumar Sharma, 2261526, Rohit Joshi S/o Mr. Harish Chandra Joshi, 2261487, Navneet Pandey S/o Mr. Neeraj Chandra Pandey, 2261384, Abhinav Singh Bohra S/o Mr. Gagan Singh, 2261063 of B.Tech.(CSE) to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. They have worked under my guidance and supervision and fulfilled the requirement for the submission of a report.

Mr. Anubhav Bewerwal
(Project Guide)

Dr. Ankur Singh Bisht
(Head, CSE)

ACKNOWLEDGEMENT

We take immense pleasure in thanking the Honorable Director **‘Prof. (Col.) Anil Nair (Retd.)’,** GEHU Bhimtal Campus to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance, and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me with everything that we need. We again want to extend thanks to our president **‘Prof. (Dr.) Kamal Ghanshala’** for providing us with all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to **‘Dr. Ankur Singh Bisht’** (Head, Department of Computer Science and Engineering, GEHU Bhimtal Campus), our project guide **‘Mr. Anubhav Bewerwal** (Assistant Professor, Department of Computer Science and Engineering, GEHU Bhimtal Campus) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this report.

Finally, yet importantly, We would like to express my heartiest thanks to our beloved parents, for their moral support, affection, and blessings. We would also like to pay our sincere thanks to all my friends and well-wishers for their help and wishes for the successful completion of this project.

Shivam Kaushik, 2261526
Rohit Joshi, 2261487
Navneet Pandey, 2261384
Abhinav Singh Bohra, 2261063

Abstract

The Hostel Management System is a web-based application developed using Python (Flask) for the backend, with HTML, CSS, and JavaScript on the frontend. It aims to streamline and automate various administrative and operational aspects of hostel management, including room booking, student allocation, complaint resolution, and payment tracking.

The project is driven by the need to replace traditional manual hostel recordkeeping with a more efficient, scalable, and accessible solution. It is built to support separate access roles for students, staff, and admin, each with distinct functionalities and privileges. Students can register, book rooms, submit complaints, and view their room allocation, while admins can manage student data, handle room requests, resolve complaints, and oversee hostel capacity.

The system integrates core concepts of database management, role-based access control, and user authentication, providing a practical understanding of full-stack application development. It includes essential features such as dynamic room availability, admin-controlled booking approval, and real-time updates, all managed through a centralized interface and a robust backend powered by SQLite.

While it does not employ advanced UI frameworks, the application delivers a smooth and intuitive user experience through its functional and clean interface. Designed primarily for educational purposes, the system encourages learning through real-world implementation of web technologies, database interactions, and modular architecture.

Looking forward, the project can be extended to include advanced features such as email notifications, hostel fee management, QR-based room entry, and mobile app integration, making it a comprehensive solution for institutional hostel administration.

TABLE OF CONTENTS

INTRODUCTION

1.1 Prologue

In the digital age where convenience meets necessity, managing operations manually has become a significant limitation, especially in the domain of student accommodation. The Hostel Booking System project has been developed to simplify and digitize the entire process of booking hostels for students, ensuring accuracy, transparency, and accessibility. This system aims to eliminate the traditional paper-based approach and offers a streamlined, centralized solution.

1.2 Background and Motivations

The conventional method of hostel booking is marred with challenges such as long queues, paperwork, miscommunication, and data mismanagement. These factors lead to inefficient allocation and dissatisfaction among students. The motivation behind developing this Hostel Booking System arises from the need to overcome these inefficiencies and build a reliable, user-friendly digital solution. Our system not only reduces the administrative burden but also provides students the ease of booking rooms from anywhere.

1.3 Problem Statement

Managing hostel room bookings manually often leads to errors, duplication, and data loss. Additionally, tracking occupancy, complaints, and payments becomes cumbersome without a centralized system. Thus, there is a need for a robust system that automates hostel booking processes and integrates necessary functionalities under a single platform.

1.4 Objectives and Research Methodology

- To design and develop an interactive web-based Hostel Booking System.
- To provide modules for student registration, room selection, complaint submission, and admin control.
- To ensure data integrity, security, and easy access to booking history.

The project was developed using the waterfall methodology. Initial requirements were gathered through discussion and research. The design was modeled using flowcharts and ER diagrams. Python (Flask framework) was used for backend development, while HTML, CSS, and JavaScript supported the frontend. SQLite served as the database system.

1.5 Project Organization

This report is organized into multiple chapters:

- **Chapter 2:** Discusses hardware and software requirements.
- **Chapter 3:** Provides implementation code and function descriptions.
- **Chapter 4:** Offers snapshots of the working application.
- **Chapter 5:** Highlights the limitations of the system.
- **Chapter 6:** Suggests enhancements for future versions.
- **Chapter 7:** Presents the overall conclusion of the project.

HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Hardware Requirement

The minimum hardware requirements for running the Hostel Booking System are:

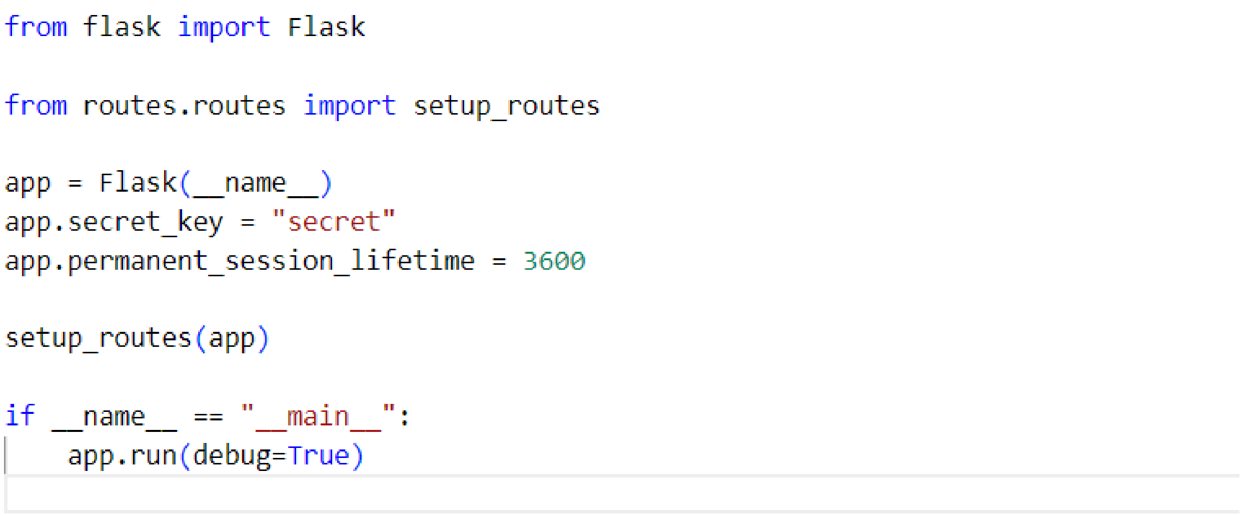
- A computer with at least a 1.5 GHz processor (Intel/AMD)
- Minimum 4 GB RAM
- Minimum 100 MB of available disk space
- Internet connectivity for accessing the web interface (optional for local testing)

2.2 Software Requirement

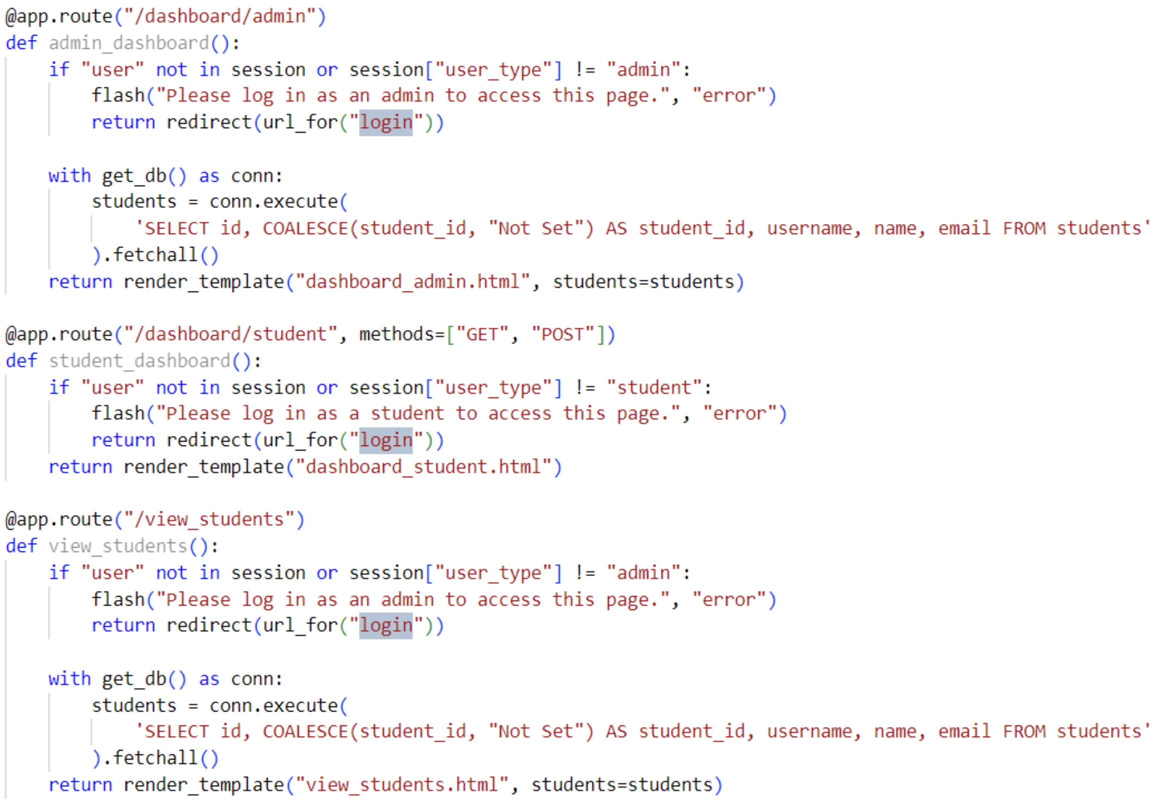
To develop and deploy the Hostel Booking System, the following software components were used:

- **Operating System:** Windows 10 or Linux-based OS
- **Backend Framework:** Python with Flask
- **Frontend Technologies:** HTML5, CSS3, JavaScript
- **Database:** SQLite3 (used for lightweight, file-based storage)
- **Text Editor/IDE:** VS Code or any Python-compatible IDE
- **Browser:** Google Chrome, Mozilla Firefox, or any modern browser

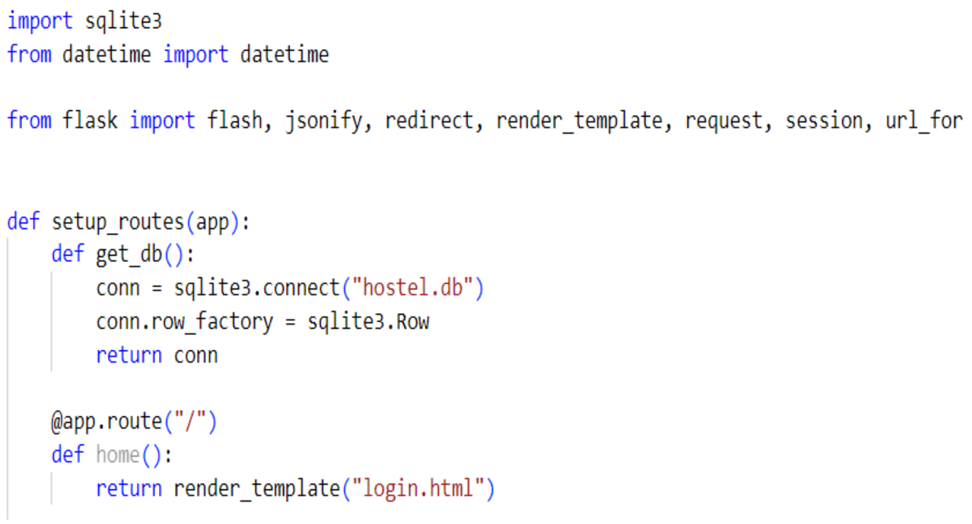
CODE:



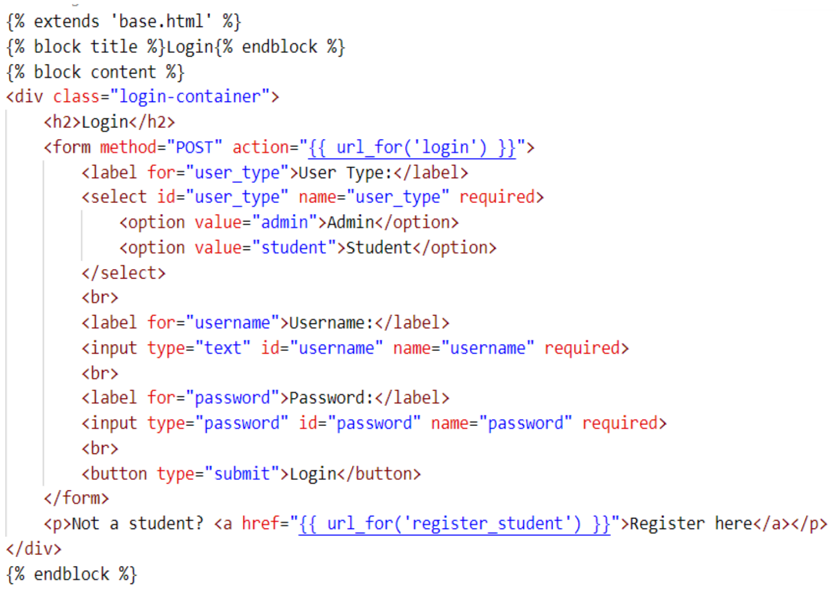
```python
from flask import Flask

from routes.routes import setup_routes

app = Flask(__name__)
app.secret_key = "secret"
app.permanent_session_lifetime = 3600

setup_routes(app)

if __name__ == "__main__":
    app.run(debug=True)
```



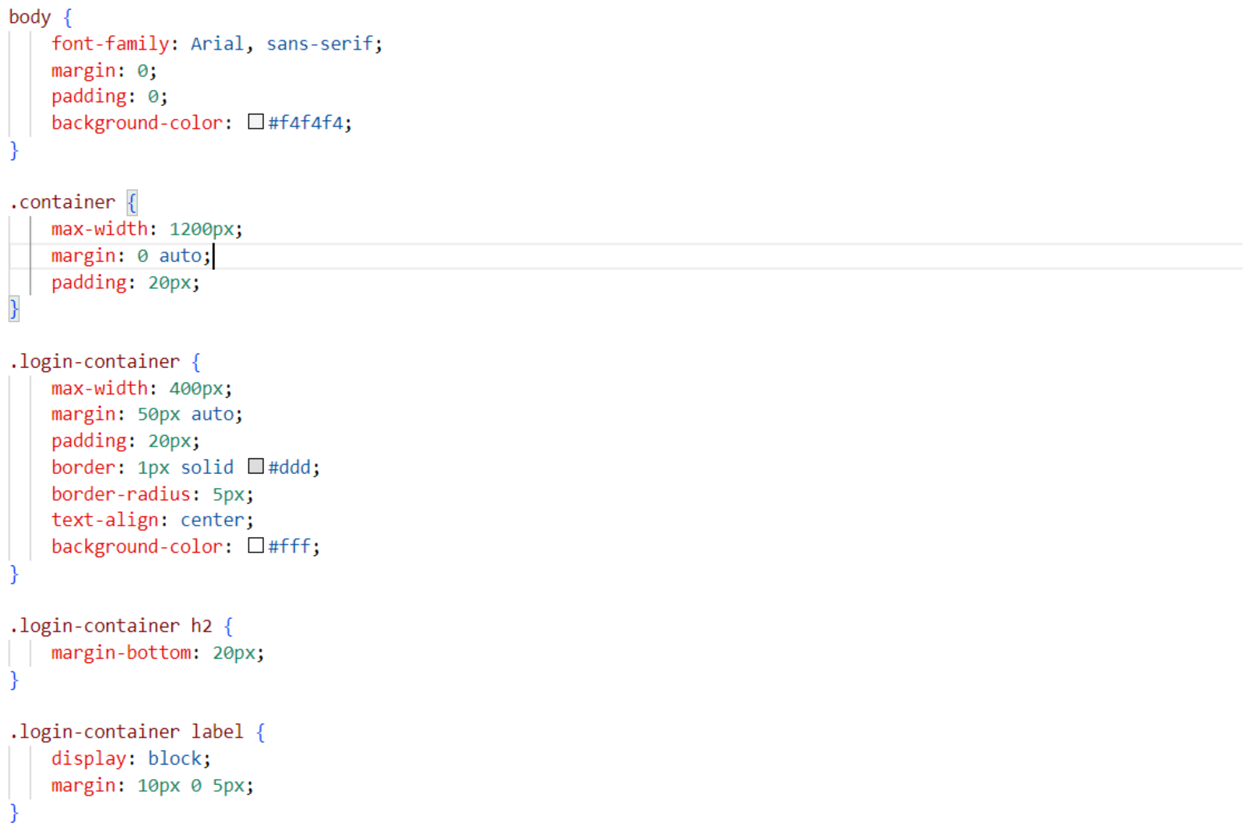
```python
@app.route("/dashboard/admin")
def admin_dashboard():
    if "user" not in session or session["user_type"] != "admin":
        flash("Please log in as an admin to access this page.", "error")
        return redirect(url_for("login"))

    with get_db() as conn:
        students = conn.execute(
            'SELECT id, COALESCE(student_id, "Not Set") AS student_id, username, name, email FROM students'
        ).fetchall()
    return render_template("dashboard_admin.html", students=students)

@app.route("/dashboard/student", methods=["GET", "POST"])
def student_dashboard():
    if "user" not in session or session["user_type"] != "student":
        flash("Please log in as a student to access this page.", "error")
        return redirect(url_for("login"))
    return render_template("dashboard_student.html")

@app.route("/view_students")
def view_students():
    if "user" not in session or session["user_type"] != "admin":
        flash("Please log in as an admin to access this page.", "error")
        return redirect(url_for("login"))

    with get_db() as conn:
        students = conn.execute(
            'SELECT id, COALESCE(student_id, "Not Set") AS student_id, username, name, email FROM students'
        ).fetchall()
    return render_template("view_students.html", students=students)
```



```python
import sqlite3
from datetime import datetime

from flask import flash, jsonify, redirect, render_template, request, session, url_for


def setup_routes(app):
    def get_db():
        conn = sqlite3.connect("hostel.db")
        conn.row_factory = sqlite3.Row
        return conn

    @app.route("/")
    def home():
        return render_template("login.html")
```

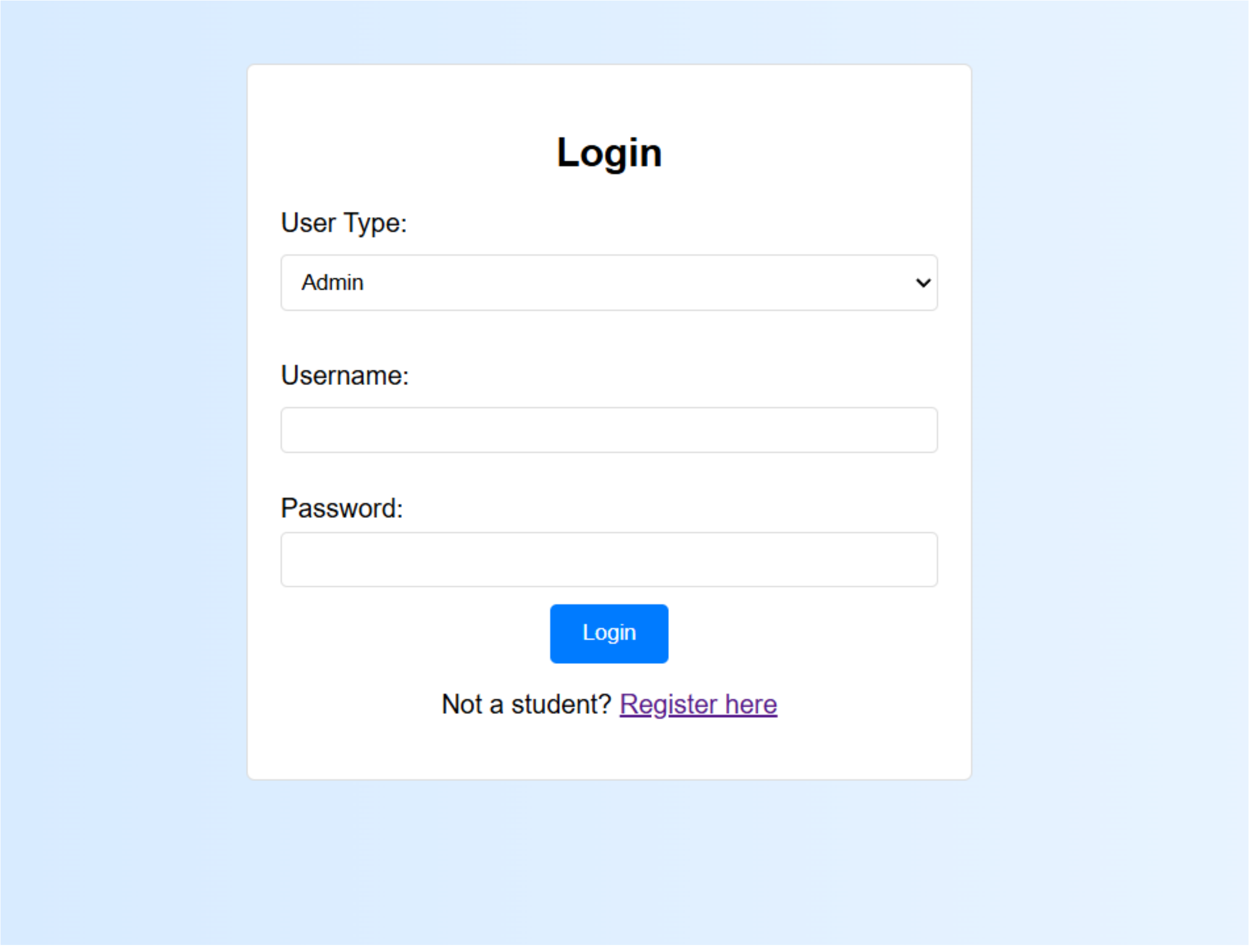


```html
{% extends 'base.html' %}
{% block title %}Login{% endblock %}
{% block content %}
<div class="login-container">
    <h2>Login</h2>
    <form method="POST" action="{{ url_for('login') }}">
        <label for="user_type">User Type:</label>
        <select id="user_type" name="user_type" required>
            <option value="admin">Admin</option>
            <option value="student">Student</option>
        </select>
        <br>
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required>
        <br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
        <br>
        <button type="submit">Login</button>
    </form>
    <p>Not a student? <a href="{{ url_for('register_student') }}">Register here</a></p>
</div>
{% endblock %}
```



```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

.container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
}

.login-container {
    max-width: 400px;
    margin: 50px auto;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 5px;
    text-align: center;
    background-color: #fff;
}

.login-container h2 {
    margin-bottom: 20px;
}

.login-container label {
    display: block;
    margin: 10px 0 5px;
}
```

SNAPSHOTS



Login

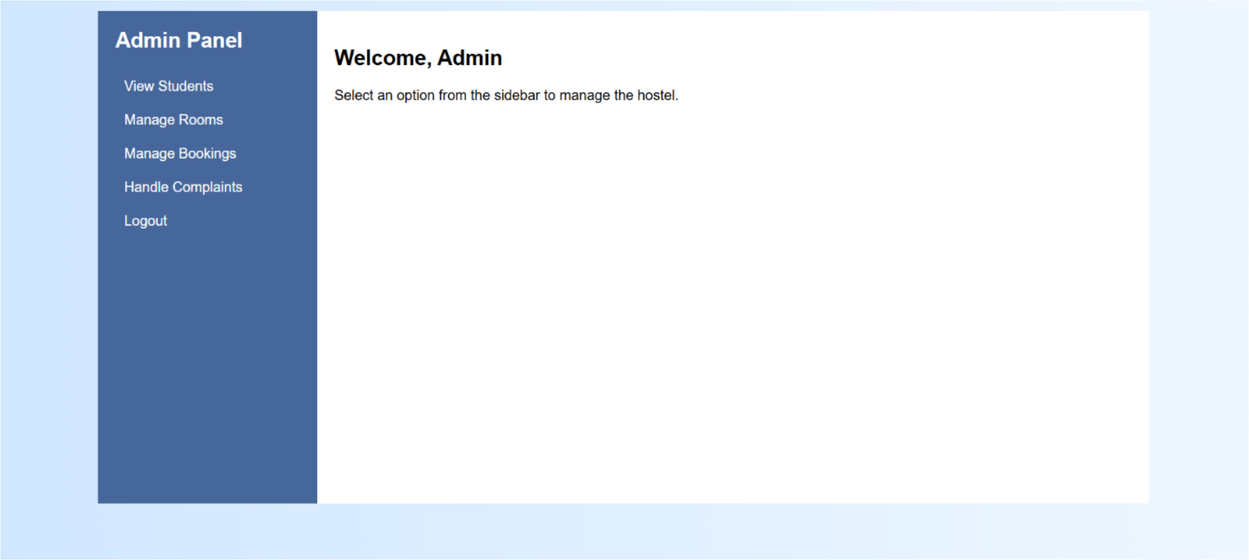
User Type:

Admin

Username:

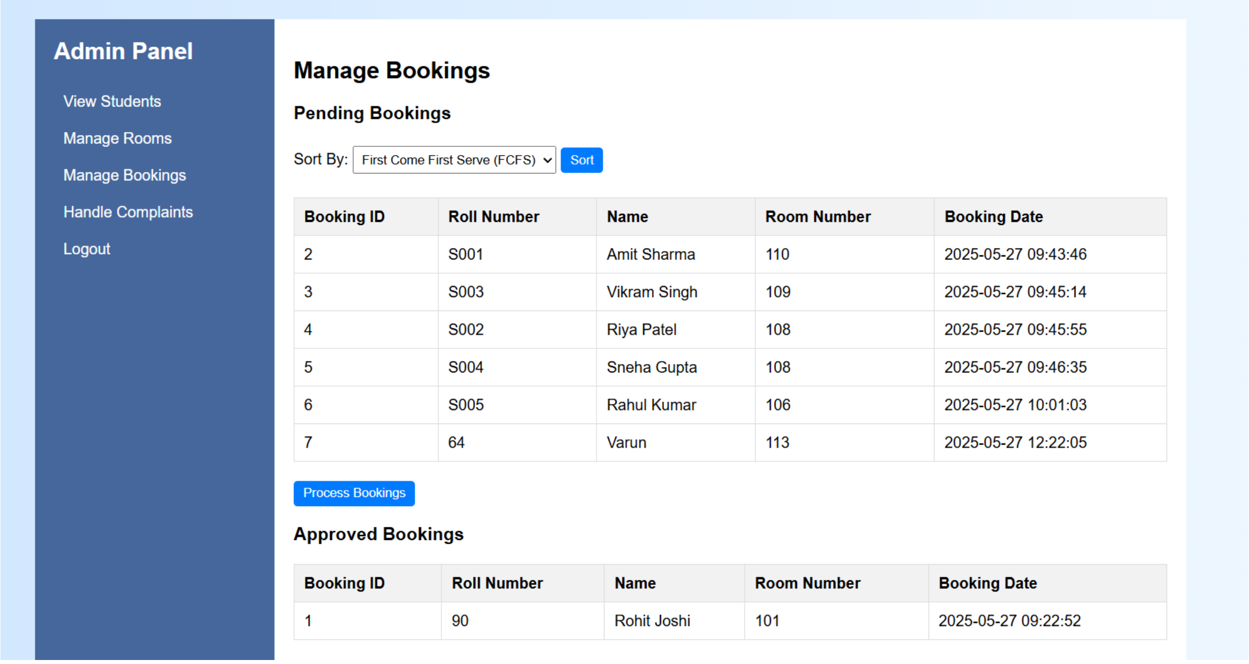
Password:

Not a student? Register here



Welcome, Admin

Select an option from the sidebar to manage the hostel.



Manage Bookings

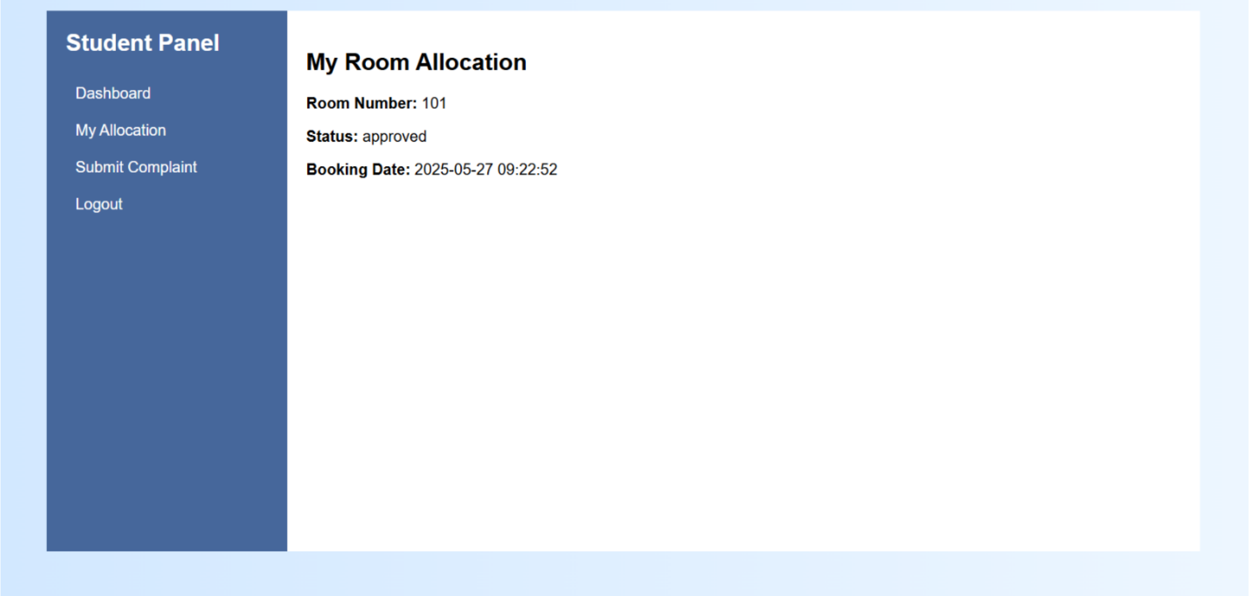
Pending Bookings

Sort By: First Come First Serve (FCFS)

Booking ID	Roll Number	Name	Room Number	Booking Date
2	S001	Amit Sharma	110	2025-05-27 09:43:46
3	S003	Vikram Singh	109	2025-05-27 09:45:14
4	S002	Riya Patel	108	2025-05-27 09:45:55
5	S004	Sneha Gupta	108	2025-05-27 09:46:35
6	S005	Rahul Kumar	106	2025-05-27 10:01:03
7	64	Varun	113	2025-05-27 12:22:05

Approved Bookings

Booking ID	Roll Number	Name	Room Number	Booking Date
1	90	Rohit Joshi	101	2025-05-27 09:22:52

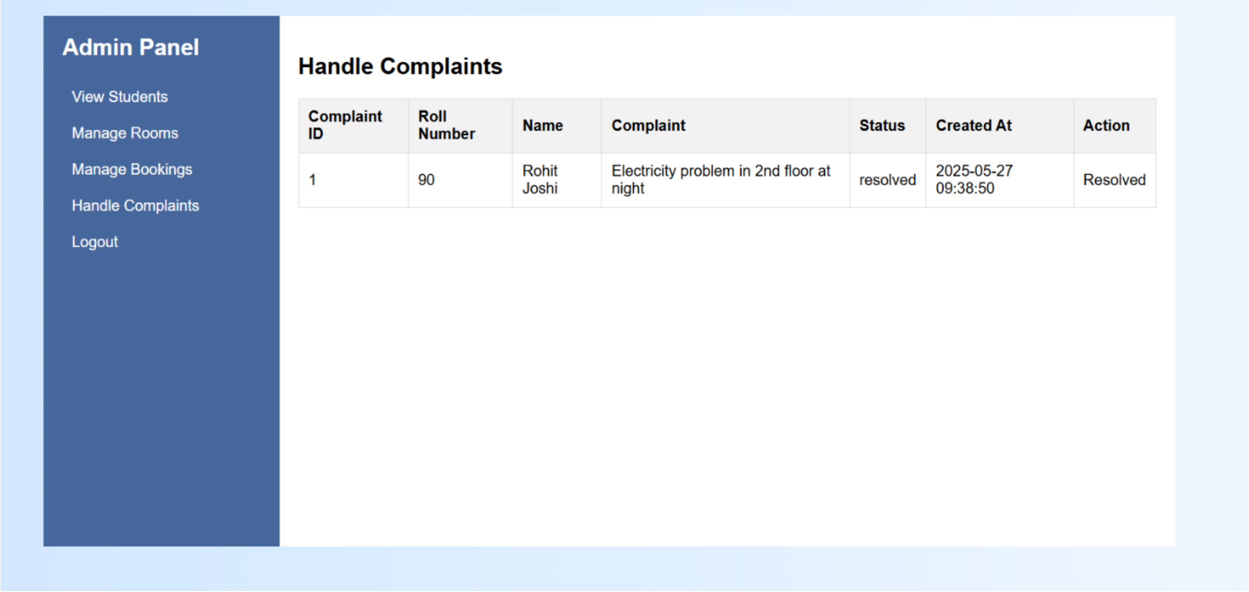


My Room Allocation

Room Number: 101

Status: approved

Booking Date: 2025-05-27 09:22:52



Handle Complaints

Complaint ID	Roll Number	Name	Complaint	Status	Created At	Action
1	90	Rohit Joshi	Electricity problem in 2nd floor at night	resolved	2025-05-27 09:38:50	Resolved

LIMITATIONS

Despite its functional strengths and core feature implementations, the *Hostel Booking System* is not without its limitations. These limitations arise primarily due to the scope of the project, time constraints, and the use of simplified development tools for learning purposes.

5.1 **Single Admin Access**

Currently, the system allows only a single hardcoded admin account for management purposes. It lacks role-based access control, which would be necessary in a multi-admin or departmental setup.

5.2 **No Real-Time Notifications**

The system does not support real-time notifications or alerts for booking confirmations, complaints, or room allocations. Users must manually check their dashboard for updates.

5.3 **Limited Scalability**

Due to the use of SQLite as the database engine, the system may not perform efficiently under heavy concurrent access or with large datasets. For institutional deployment, a more robust database such as MySQL or PostgreSQL would be ideal.

5.4 **No Email or SMS Integration**

There is no integration with external services like email or SMS gateways to notify users of important updates or booking confirmations.

5.5 **Basic UI and UX Design**

While the interface is functional, it lacks modern UI/UX enhancements. Features such as responsive design, animations, and accessibility features are limited.

ENHANCEMENTS

Although the *Hostel Booking System* fulfills its core objectives, there are several enhancements that can be implemented to elevate its functionality, user experience, and scalability. These enhancements will also help in aligning the system with real-world deployment standards.

Email and SMS Notification System

Integrating email or SMS services can automate notifications to students regarding booking status, complaint responses, and room assignments. This would improve communication and reduce the need for manual follow-ups.

Responsive and Modern UI Design

Redesigning the user interface using modern frameworks like Bootstrap or Tailwind CSS can enhance accessibility across devices and improve the overall user experience. Including animations, modals, and AJAX calls can also provide seamless interactivity.

Real-Time Data Updates with WebSockets

To enable real-time booking status updates and instant complaint notifications, WebSockets or polling-based systems can be implemented.

Advanced Complaint Management System

A more sophisticated complaint module with categories, priorities, status tracking (open, in-progress, resolved), and feedback mechanisms would streamline resolution and accountability.

Search and Filter Features

Adding options to filter and search for rooms based on capacity, status, and amenities would make the booking process more efficient for students.

Reports and Analytics Dashboard

An admin dashboard that displays statistics such as total bookings, room availability trends, and complaint logs can provide valuable insights and aid in decision-making.

CONCLUSION

The *Hostel Booking System* represents a significant step toward digitalizing and streamlining the hostel room allocation process for students. By integrating features such as student login/registration, room request handling, and administrative oversight through a centralized web interface, the system provides a user-friendly and efficient alternative to manual allocation processes.

Through this project, the core principles of full-stack web development, including frontend design, backend integration, and database management, were practically implemented. It also offered valuable exposure to software development cycles, real-time problem-solving, and modular system architecture.

The application currently facilitates room booking requests, student complaints, and admin responses in an organized and accessible manner. Though the current implementation is functional and caters to primary needs, it leaves open avenues for several enhancements such as automation, real-time updates, and greater system scalability.

In conclusion, the *Hostel Booking System* not only serves as a working software solution but also stands as a testament to the practical application of theoretical knowledge gained in computer science engineering. With further refinement, this project has the potential to evolve into a robust platform for institutional hostel management across campuses.

REFERENCES

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th Edition). Wiley.
2. Flask Documentation
3. W3Schools HTML, CSS, and JavaScript Tutorials
4. SQLite Official Documentation
5. MDN Web Docs
6. Stack Overflow –For community-based programming support and debugging
7. VS Code Documentation
8. Real Python – Tutorials and guides on Python development