RESEARCH ARTICLE OPEN ACCESS

**Preparation of Papers for International Journal of Scientific Research and Engineering Development**

**"Towards Digital Hostel Administration: A Modular Web Application Approach"**

Yashodha Sambrani, Pooja Shekharagouda Doddagoudar, Smital Sanjay Kaginkar,

Punarvasu Shetake, Ambika Thabbasa Chavan

(Computer Science Department, Shri Dharmasthala Manjunatheshwar College of Engineering and Technology (SDMCET), Dharwad. Email: [ysambrani@gmail.com](mailto:ysambrani@gmail.com), [poojasdoddagoudar@gmail.com](mailto:poojasdoddagoudar@gmail.com), [kaginkarsmital@gmail.com](mailto:kaginkarsmital@gmail.com), [shetakepunarvasu@gmail.com](mailto:shetakepunarvasu@gmail.com), [chavanambika088@gmail.com](mailto:chavanambika088@gmail.com). )

----------------------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*----------------------------------

Abstract:

This document describes a web-based Hostel Management System designed to simplify and optimize the management of student accommodations. The system, built with Flask for the backend, React for the frontend, and MongoDB for data storage, offers secure registration for students, dynamic allocation of rooms, management of leave requests, notifications, and administrative controls. The application enhances operational efficiency, minimizes paperwork, and improves accessibility for both students and administrators.

Keywords **—Hostel management, Flask, HTML, CSS, JS, MongoDB, room allocation, leave management, student portal, admin dashboard.**

----------------------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*----------------------------------

1. **INTRODUCTION**

In educational institutions, running dorms can be a challenging undertaking. This has historically required manual procedures including keeping paper records of student information, managing room assignments, processing leave requests using written forms, and disseminating notices via printed circulars or word-of-mouth. These traditional approaches are ineffective, time-consuming, and prone to mistakes; they frequently result in data loss and communication delays.

In order to overcome these obstacles, we created a web-based hostel management system that uses a centralized, user-friendly platform to automate the entire process. The system offers dependable speed and smooth interaction because it was constructed with Flask for backend API functionality, HTML, CSS and JS for a dynamic frontend experience, and MongoDB for flexible data storage.

Notifications, room management, leave processing, student registration and login, and an admin dashboard are some of the main features. To guarantee seamless operation for administrators and students alike, every module is integrated.

The solution improves efficiency, guarantees real-time data access, and drastically lowers administrative strain by digitizing hostel operations, thus providing a better experience for all users.

1. **SYSTEM DESIGN AND ARCHITECTURE**

The technologies used and the modular architecture used in the Hostel Management System's development are described in this section. With a distinct division between the frontend, backend, and database layers, the system is organized as a web application. Scalability, maintainability, and the simplicity of future improvements are guaranteed by this modular design.

1. **Tools and Technologies Used**

To build a responsive, secure, and dynamic web-based hostel management system, the following tools and technologies were used:

* Frontend:

HTML, CSS, and JavaScript form the foundational technologies used to build the frontend of the Hostel Management System. HTML provides the structure of the web pages, CSS ensures responsive and visually appealing design, and JavaScript enables interactivity and dynamic behaviour on the client side. Together, they deliver a smooth and user-friendly interface for both students and administrators, enhancing usability and accessibility across all modules.

* Backend: Flask (Python)

Flask is a lightweight and adaptable Python web framework utilized for developing RESTful APIs. It manages backend logic, processes HTTP requests, and oversees business rules such as room allocation, marks attendance, notifies updates and complains and leave approval.

* Database: MongoDB

MongoDB, a NoSQL document-oriented database, is employed to store and manage both structured and semi-structured data. Its inherent flexibility allows for straightforward schema modifications, making it well-suited for managing diverse data types such as student profiles, room records, attendance marked, complains registered and leave applications.

1. **System Modules**
   1. **Student Registration and Login Module**

* New students can register by entering their personal and academic details.
* Profile picture uploads are supported.
* Passwords are securely stored, and login.
  1. **Room Management Module**
* Admins can add, remove, and update room information.
* Rooms can be assigned to or freed from students dynamically.
* Room occupancy status is continuously tracked and displayed.
  1. **Leave Management System**
* Students submit leave applications specifying duration and reason.
* Admins review and approve or reject requests with optional comments.
* Leave history is maintained for each student.
  1. **Attendance Management**
* Students can mark daily attendance on the given time.
* Attendance records are time-stamped and stored for future reference.
* Reports can be generated to identify present students.
  1. **Notification System**
* Admins can post announcements visible to all users.
* Notifications include alerts for holidays, maintenance, events, and updates.
* Students receive real-time updates on home dashboard.
  1. **Complaints Handling System**
* Students can submit complaints related to facilities, safety, or other issues.
* Each complaint includes a title, description and category.
* Admins view the status of complaints.
  1. **Admin Dashboard**
* Centralized control panel to monitor all modules.
* Overview of room status, student registrations, attendance, complaints and notifications.
* Interfaces for managing all database collections through user-friendly controls.

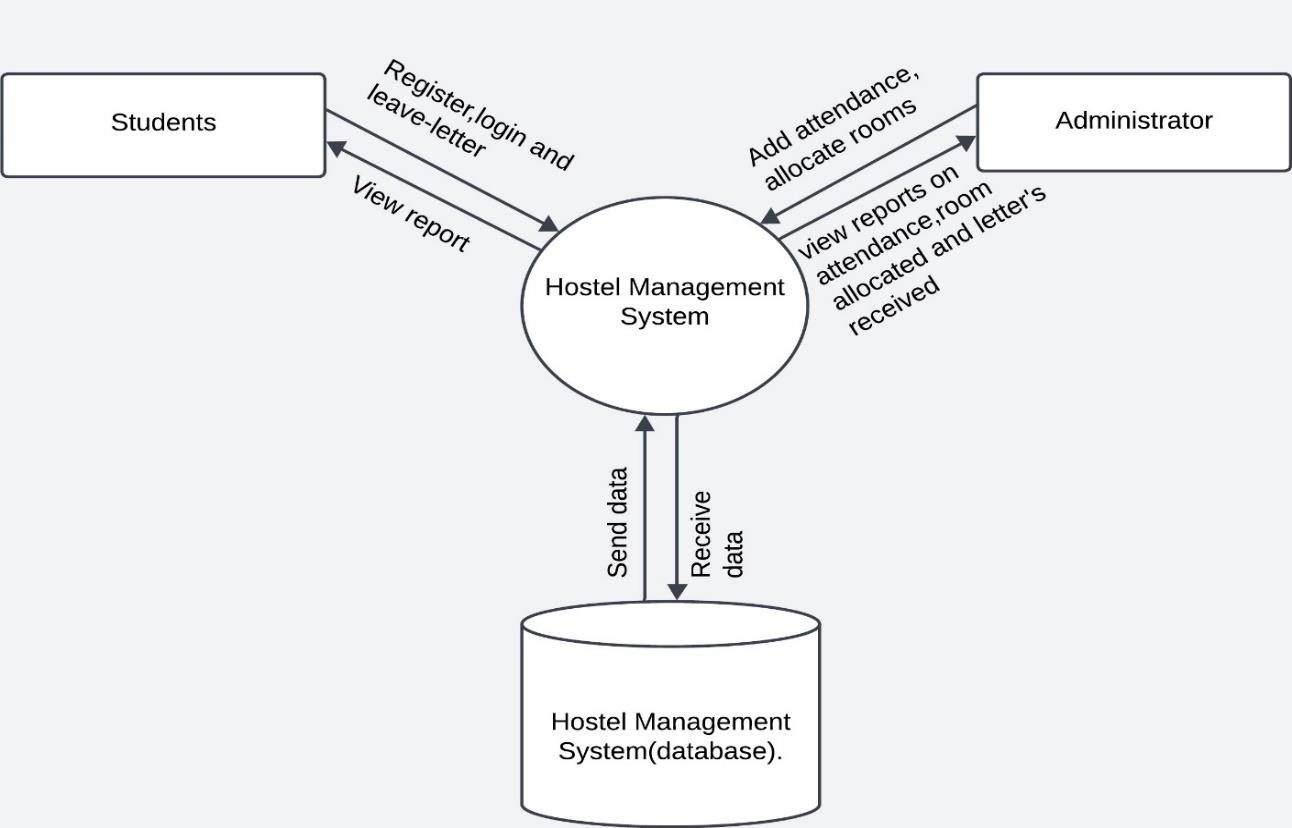


Fig1: System architecture

**UML DIAGRAMS**

**1. Use - Case Diagram:**

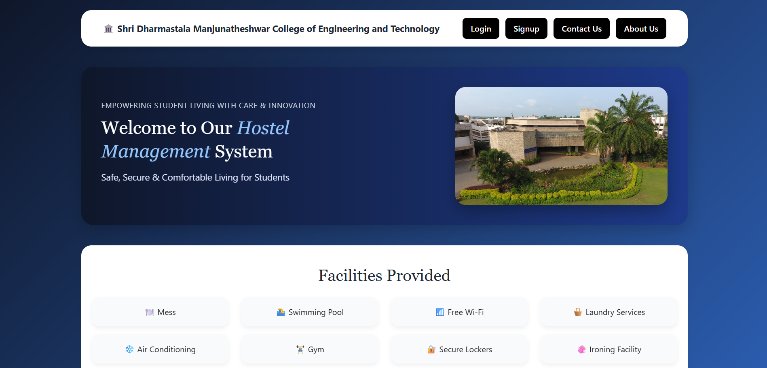
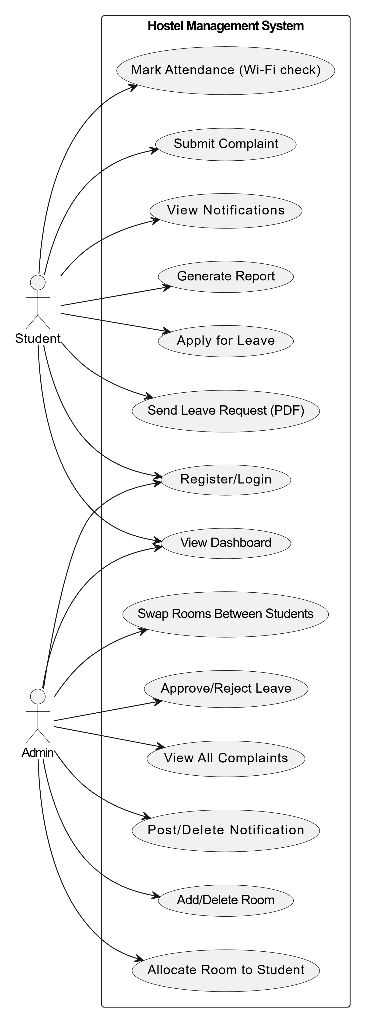
****

Fig2: Use - Case diagram

**2.Activity Diagram**

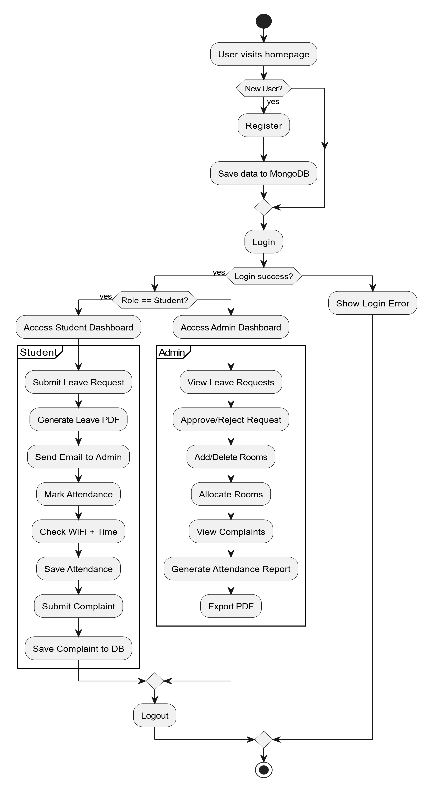
****

Fig3: Activity Diagram

**3.** **Class Diagram:**

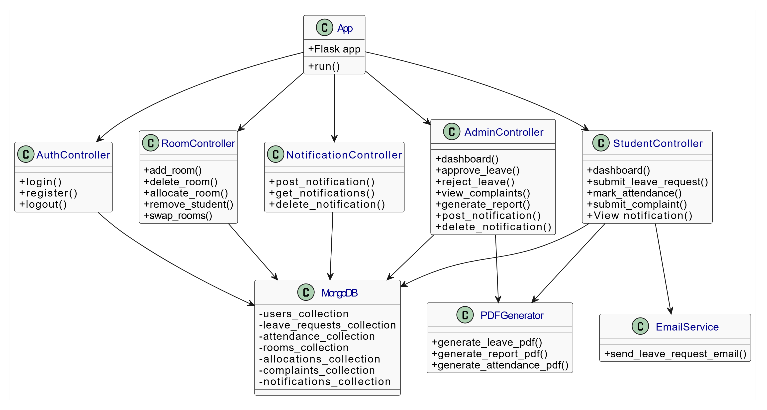
****

Fig4: Class Diagram

1. **RESULTS AND DISCUSSION**

The Hostel Management System was successfully developed and tested across various modules including student registration, room allocation, leave management, attendance tracking, notifications, and complaint handling. The system was evaluated for both functionality and user experience.

* 1. ***Features of the Hostel Management System***

1. Digital Attendance for Students where they can mark their attendance through the system using their personal

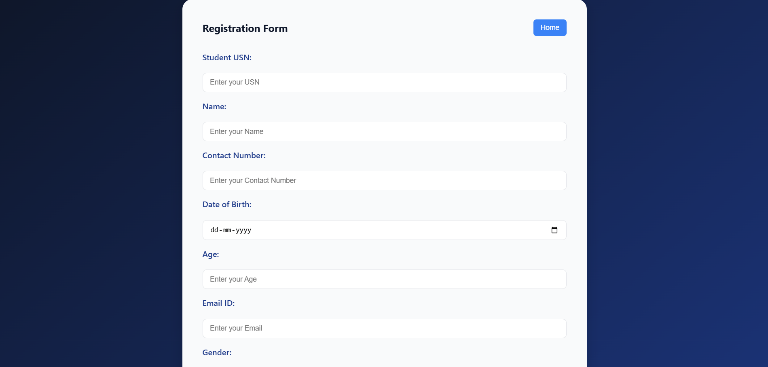
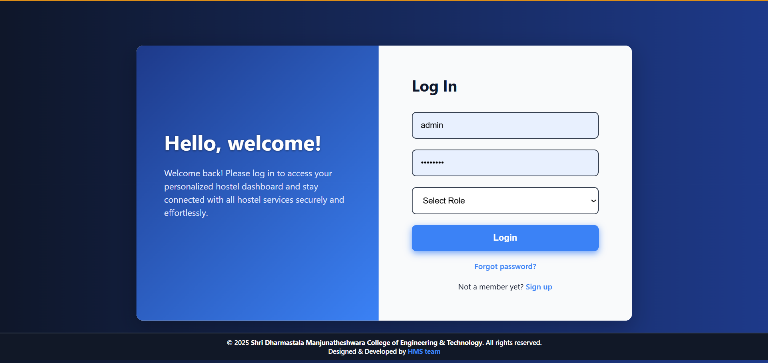
Fig5: Home page

Fig6: Registration page

Fig7: Login page

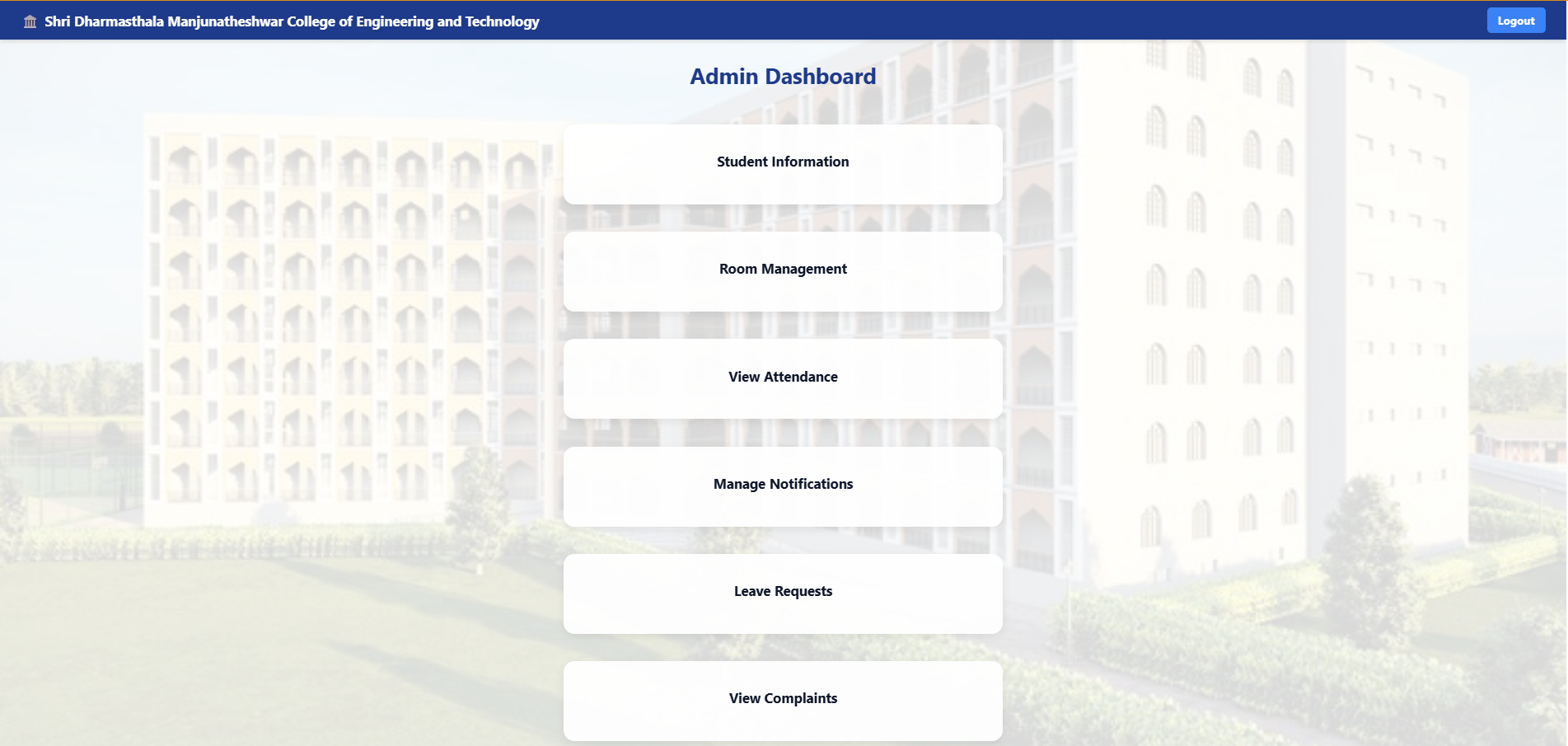


Fig8: Admin Dashboard

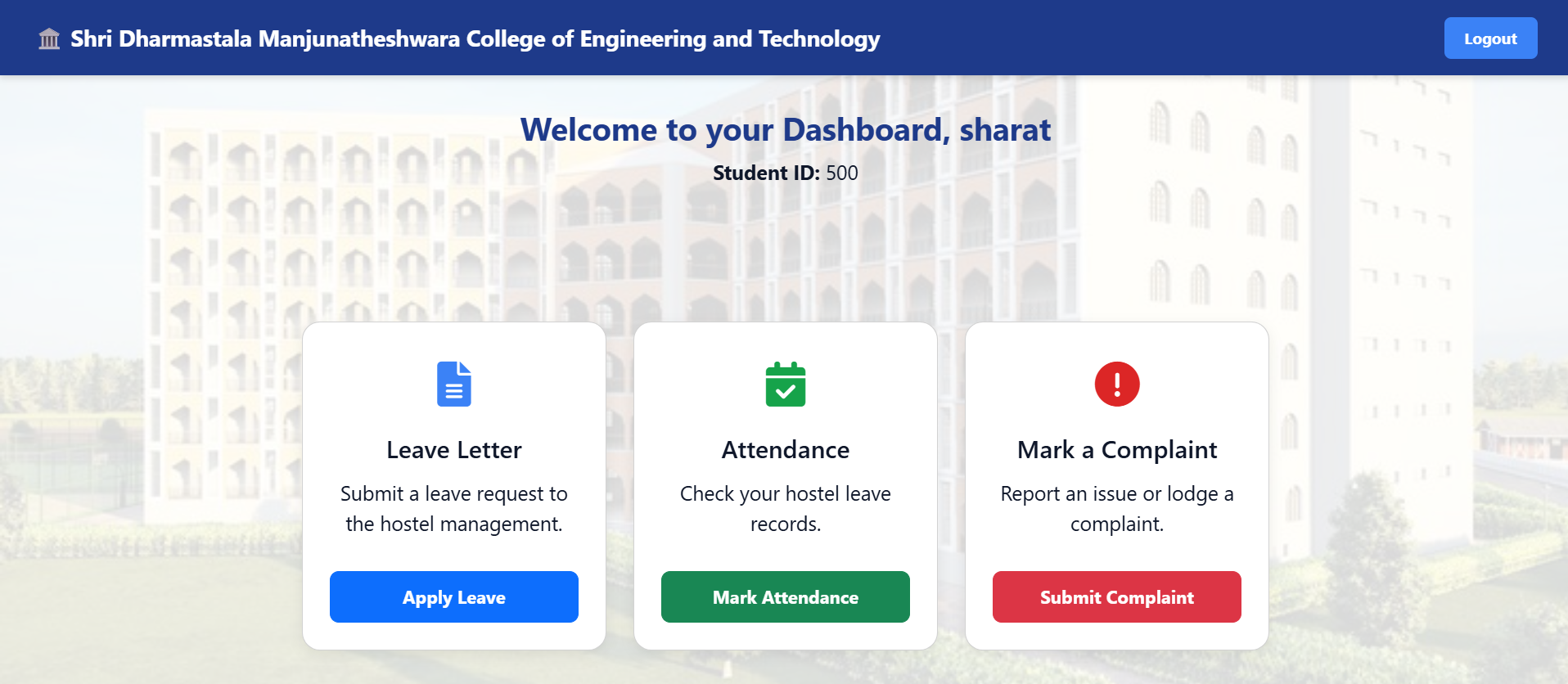


Fig9: Student Dashboard

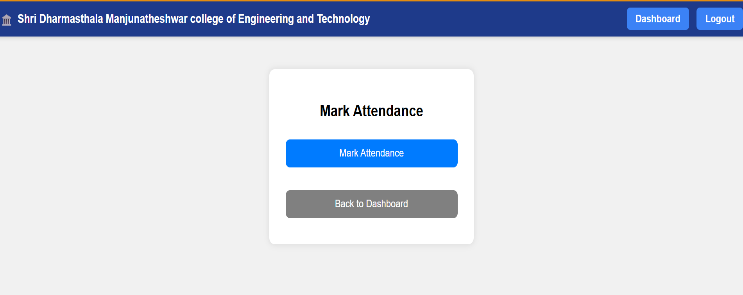
1. Real-Time Attendance Monitoring   
   Administrators can track and monitor student attendance as it happens.

Fig10: Mark attendance page(student)

1. Automated Daily Attendance Reports  
   The system generates daily attendance reports automatically for review and records.

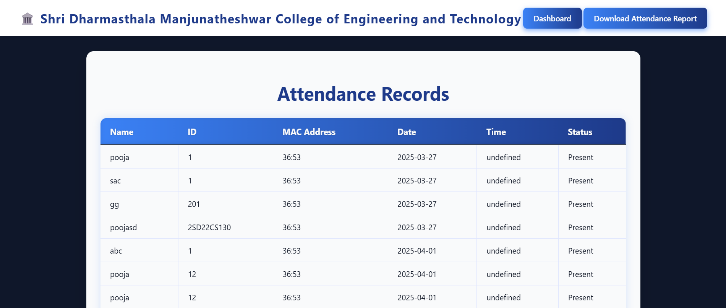


Fig11: view attendance(admin)

1. Room Allocation Based on Availability  
   Rooms are assigned to students dynamically, based on current room availability.

Fig12: Room Dashboard(admin)

1. Room Change Request Handling  
   Students can submit room change requests, which admins can approve or reject through the system.



Fig13: Room allocation (admin)

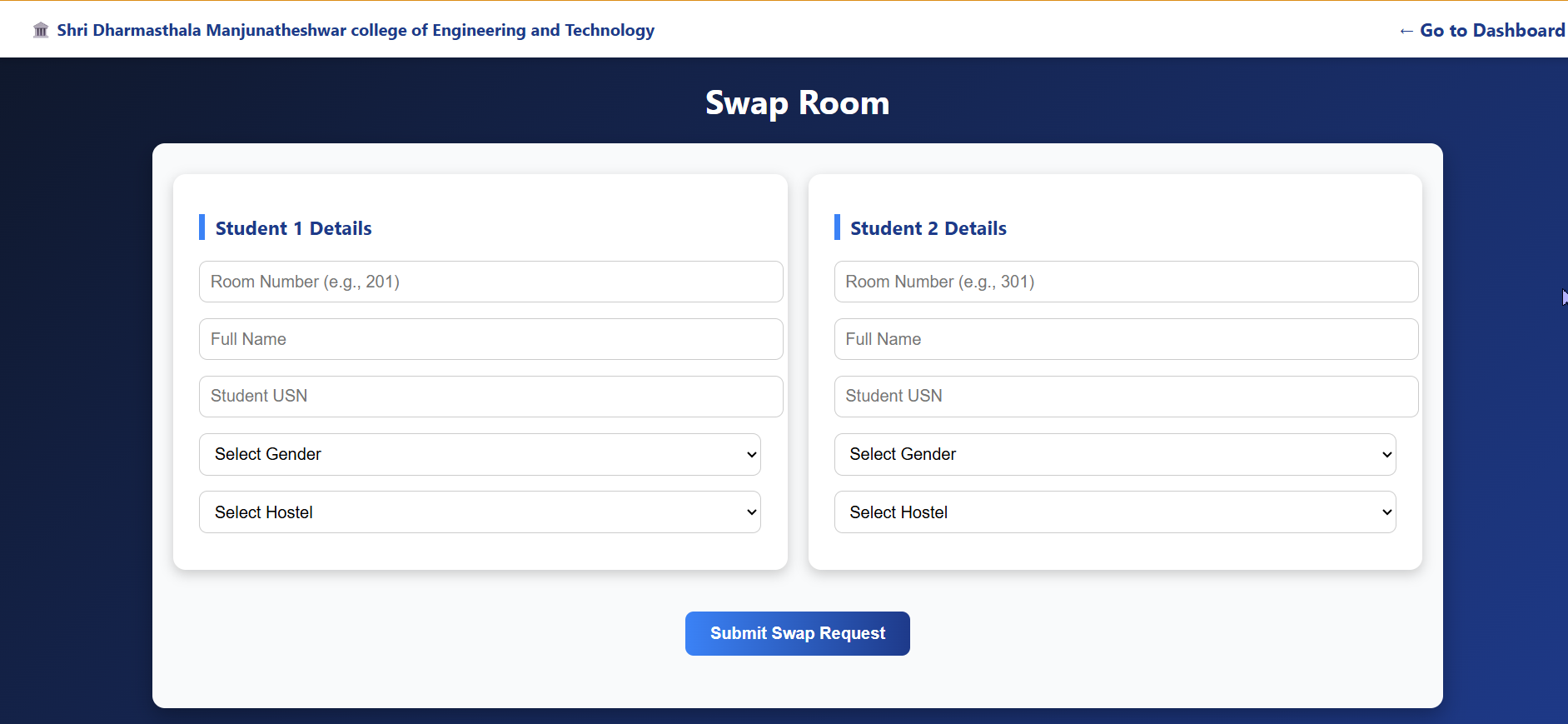


Fig14: Swap rooms (admin)

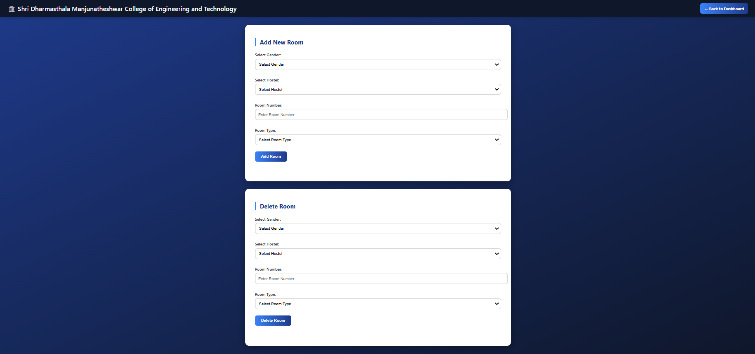


Fig15: Manage rooms – add and delete (admin)

1. Notifications are sent to students by administrators regarding:

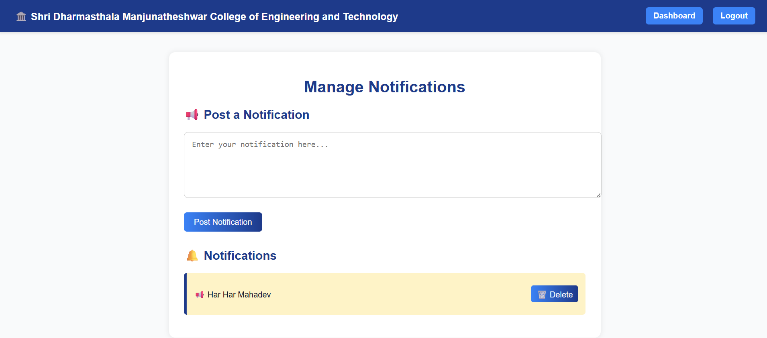
* Approval statuses
* Attendance issues
* Important announcements

Fig16: Notification page(admin)

1. Students can submit complaints about maintenance, food quality, noise, cleanliness, and other issues. Administrators can view and delete these complaints through the dashboard.

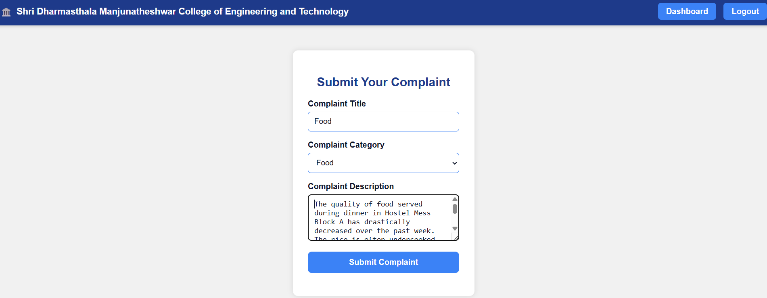
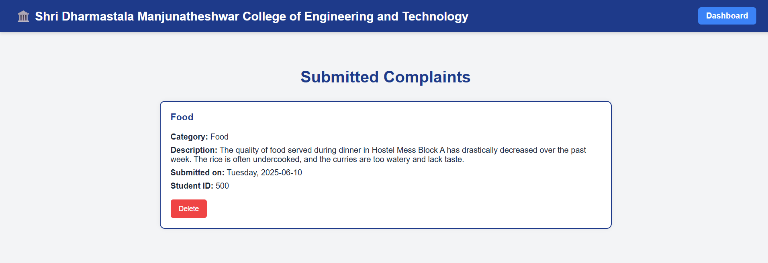
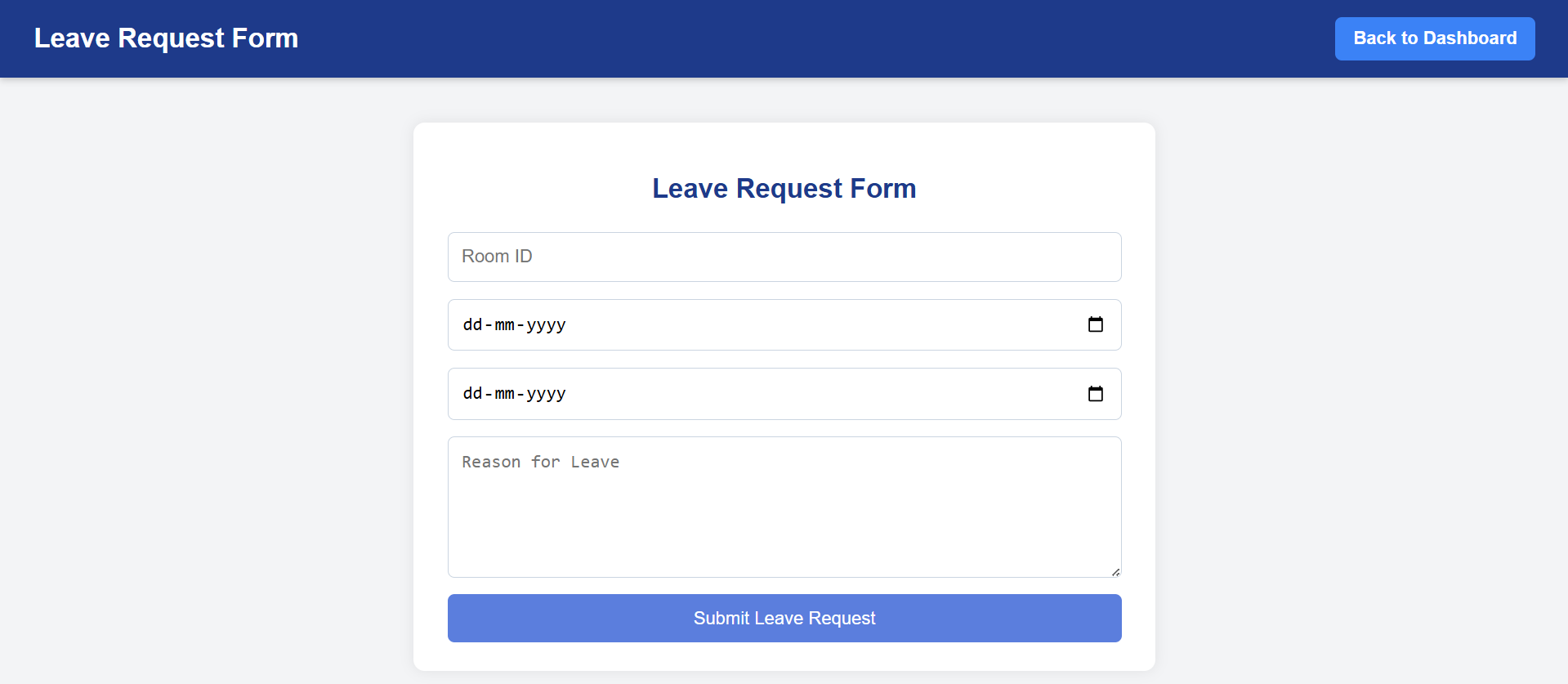


Fig17: Submit and view complaint page

8. Responsive student leave system with form submission, validation, profile display, PDF download, and leave tracking where admin approves or rejects requests.



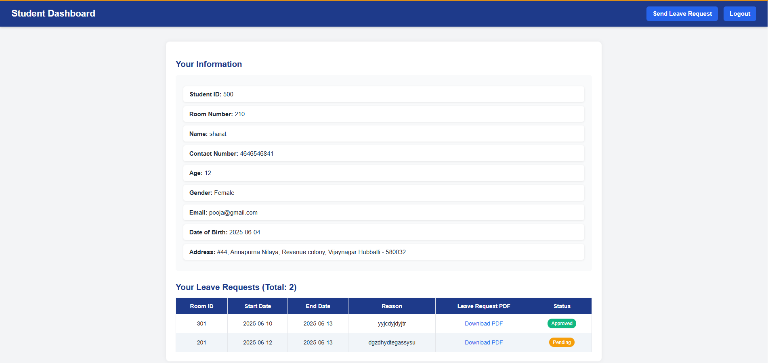
 Fig18: Leave request(student)

 Fig19: Approved and rejected status(student)

Fig20: leave request approval or rejection(admin)

**The complete deployed project link: (used Render - modern cloud platform)** <https://minorprojecthms.onrender.com>

1. **CONCLUSION**

This project presents a robust and efficient Hostel Management System built with modern web technologies, offering real-time data accessibility, streamlined hostel operations, and significantly reduced reliance on manual processes. By centralizing core functionalities such as student registration, room allocation, attendance tracking, leave management, and complaint handling, the system enhances administrative oversight, ensures data integrity, and delivers a seamless user experience for both students and hostel administrators.

**ACKNOWLEDGMENT**

We sincerely thank the faculty and staff of the Department of Computer Science and Engineering, Shri Dharmasthala Manjunatheshwar College of Engineering and Technology (SDMCET), Dharwad, for their valuable guidance, continuous support, and encouragement throughout the development of this project. We are also grateful for the resources and environment provided by the institution that enabled us to successfully implement and test our system.

1. **REFERENCES**

[1] Flask Documentation-

[*https://flask.palletsprojects.com/*](https://flask.palletsprojects.com/)  
[2] ReactJS Official Documentation – [*https://reactjs.org/*](https://reactjs.org/)  
[3] MongoDB Manual – [*https://www.mongodb.com/docs/manual/*](https://www.mongodb.com/docs/manual/)  
[4] Bootstrap Official Site – [*https://getbootstrap.com/*](https://getbootstrap.com/)  
[5] Postman API Testing Tool – [*https://www.postman.com/*](https://www.postman.com/)  
[6] Axios GitHub Repository – [*https://github.com/axios/axios/*](https://github.com/axios/axios/)  
[7] JWT.IO - JSON Web Tokens Introduction – [*https://jwt.io/introduction/*](https://jwt.io/introduction/)  
[8] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed., O'Reilly Media, 2018.  
[9] S. Banks and E. Porcello, *Learning React*, 2nd ed., O’Reilly Media, 2020.  
[10] MongoDB Inc., *MongoDB: The Definitive Guide*, 3rd ed., O'Reilly Media, 2019.