

**A**

***PROJECT REPORT***

*on*

**Hostel Hub**

*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**

Session: - 2024

Under Guidance of

Mr. Aaditya Maheshwari

Assistant Professor CSE

Dept. of CSE TINJRIT, Udaipur

Submitted by

Krishna Agrawal (20ETCCS065)

Patel Dharmik (20ETCCS085)

Sunil Kumawat (20ETCCS107)

Surajmal Suthar (20ETCCS108)

Patel Sakshee (20ETCCS086)

8th Sem (CSE)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001 2024**

**A**

***PROJECT REPORT***

*on*

**Hostel Hub**

*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**



**Session: 2024**

Under Guidance of

Mr. Aaditya Maheshwari

Assistant Professor CSE

Dept. of CSE TINJRIT, Udaipur

Submitted by

Krishna Agrawal (20ETCCS065)

Patel Dharmik (20ETCCS085)

Sunil Kumawat (20ETCCS107)

Surajmal Suthar (20ETCCS108)

Patel Sakshee (20ETCCS086)

8th Sem (CSE)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001**



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled **Hostel Hub** by **Krishna Agrawal** was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Mr. Aaditya Maheshwari

Dr. Rimpy Bishnoi

Assistant Professor CSE

Professor and HoD

Dept. of CSE TINJRIT,Udaipur

Dept. of CSET INJRIT,Udaipur

Date

Date

i



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled **Hostel Hub** by **Patel Dharmik** was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Mr. Aaditya Maheshwari

Dr. Rimpy Bishnoi

Assistant Professor CSE

Professor and HoD

Dept. of CSE TINJRIT,Udaipur

Dept. of CSET INJRIT,Udaipur

Date

Date

ii



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled **Hostel Hub** by **Sunil Kumawat** was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Mr. Aaditya Maheshwari

Dr. Rimpy Bishnoi

Assistant Professor CSE

Professor and HoD

Dept. of CSE TINJRIT,Udaipur

Dept. of CSET INJRIT,Udaipur

Date

Date

iii



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled **Hostel Hub** by **Surajmal Suthar** was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Mr. Aaditya Maheshwari Dr. Rimpy Bishnoi

Assistant Professor CSE Professor and HoD

Dept. of CSE TINJRIT,Udaipur Dept. of CSET INJRIT,Udaipur

Date Date

iv



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled **Hostel Hub** by **Patel Sakshee** was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Mr. Aaditya Maheshwari

Dr. Rimpy Bishnoi

Assistant Professor CSE

Professor and HoD

Dept. of CSE TINJRIT,Udaipur

Dept. of CSET INJRIT,Udaipur

Date

Date

v



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Examiner Certificate**

This is to certify that the following student

**Krishna Agrawal, Dharmik Patel, Sunil Kumawat, Surajmal Suthar, Patel Sakshee**

of final year B.Tech. (Computer Science and Engineering), was examined for the project work titled

***Hostel Hub***

during the academic year 2023 – 2024 at Techno India NJR Institute of Technology, Udaipur

**Remarks:**

**Date:**

Signature

Signature

(**Internal Examiner**)

(**External Examiner**)

Name :-

Name :-

Designation:-

Designation:-

Department: -

Department: -

Organization:-

Organization:-

vi

**ACKNOWLEDGMENT**

We take this opportunity to record our sincere thanks to all who helped us to successfully complete this work. Firstly, We are grateful to our **supervisor Mr. Aaditya Maheshwari**

for his invaluable guidance and constant encouragement, support and most importantly for giving us the opportunity to carry out this work.

We would like to express our deepest sense of gratitude and humble regards to our

**Head of Department Dr. Rimpy Bishnoi** for giving invariable encouragement in our endeavors and providing necessary facility for the same. Also a sincere thanks to all faculty members of CSE, TINJRIT for their help in the project directly or indirectly.

Finally, We would like to thank my friends for their support and discussions that have proved very valuable for us. We are indebted to our parents for providing constant support, love and encouragement. We thank them for the sacrifices they made so that we could grow up in a learning environment. They have always stood by us in everything we have done, providing constant support, encouragement and love

**Krishna Agrawal (20ETCCS065)**

**Patel Dharmik (20ETCCS085)**

**Sunil Kumawat (20ETCCS107)**

**Surajmal Suthar (20ETCCS108)**

**Patel Sakshee (20ETCCS086)**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001

vii

**CONTENTS**

**Certificate** …………………………………………………...................................

**Examiner Certificate** ……………………………………………………………….

**Acknowledgment** ………………………………………………………………..

**List of Figures** ………………………………………………….............................

i – v

vi

vii

xi

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter 1** INTRODUCTION…………………………………………................. | | 1 - 4 | 4 |
| 1.1 Name of the project ………….………………………………….. | | 1 |  |
| 1.2 | Problem Statement ……………………………………………… | 1 |  |
| 1.3 | Objective and scope of the project ……………………………. | 1 |  |
| 1.4 | Methodology .………………………………………….............. | 3 |  |
| **Chapter 2** SYSTEM ANALYSIS ………………………………………………… | | 5 - 6 |  |
| **Chapter 3** REQUIREMENTS ANALYSIS .…………………………………... | | 7-10 |  |
|  |  |  |
| 3.1 | Student Information Module ………………………………… | 7 |  |
|  |  |
| 3.1.1 Functional Requirements ………………………………….. | | 7 |  |
| 3.1.1 Non-Functional Requirements ……………………………. | | 7 |  |
| 3.2 | Complaint Resolution Module …..…………………………. | 7 |  |
| 3.2.1 Functional Requirements …………………………………. | | 7 |  |
| 3.2.2 Non-Functional Requirements …………………………… | | 8 |  |
| 3.3 | Room Allocation Module ……………………………………. | 8 |  |

viii

|  |  |  |
| --- | --- | --- |
| 3.3.1 Functional Requirements …………………………………... | | 8 |
| 3.3.2 | Non-Functional Requirements …………………………….. | 8 |
| 3.4 Notice Management Module .………………………………… | | 9 |
| 3.4.1 Functional Requirements …………………………………… | | 9 |
| 3.4.2 | Non-Functional Requirements …………………………….. | 9 |
| 3.5 Canteen Menu Management Module ……………………….. | | 10 |
| 3.5.1 Functional Requirements …………………………………… | | 10 |
| 3.5.2 | Non-Functional Requirements ……………………………… | 10 |

|  |  |
| --- | --- |
| **Chapter 4** USER PANEL ………………...……………………………………… | 11 - 13 |
| 4.1 Student Panel ……………………………………………………. | 11 |
| 4.2 Admin Panel …………………..…………………………………. | 12 |

|  |  |  |
| --- | --- | --- |
| **Chapter 5** DESINING ………………………...…..……………………………… | | 14 -19 |
| 5.1 | User Flow Diagram ………………………………………………. | 14 |
| 5.2 | Admin Flow Diagram …………….…………………….………… | 14 |
| 5.3 | UML Diagram ……………..……………………………………… | 15 |
| 5.4 | ER Diagram …………………..……………………….…………. | 16 |
| 5.5 | Data Flow Diagram ………………. …………………………….. | 17 |

|  |  |
| --- | --- |
| **Chapter 6** USER INTERFACE ……………….……………………….…………. | 21 - 26 |

|  |  |
| --- | --- |
| **Chapter 7** LIMITATION AND FUTURE OF THE PROJECT ……….………… | 27 - 30 |
| 7.1 Limitations of the project ………..……………………………… | 27 |

ix

|  |  |  |  |
| --- | --- | --- | --- |
| 7.2 | Future of the project …………….……………………………… | | 28 |
| 7.3 | Resources of the project ……………………………...………. | | 29 |
| 7.3.1 | | Software ……………………..……………………….………. | 29 |
| 7.3.2 | | Hardware …………………………. …………………………. | 29 |

**Conclusion** ….…………………………………………….....................................

**Appendices** ….……………………………………………..........................…….

**Bibliography** ….…………………………………..............................

31

32- 36

37

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Fig 5-1 | User Flow diagram …………………………………………....... | 6 |
| Fig 5-2 | Admin Flow diagram ..………………………………………….. | 6 |
| Fig 5-3 | UML diagram ………………..………………………………….. | 7 |
| Fig 5-4 | ER diagram ………...…………………………………………… | 7 |
| Fig 5-5 | Data Flow Diagram 1….……………………………………….. | 9 |
| Fig 5-6 | Data Flow Diagram 2 …………………………………………. | 10 |
| Fig 6-1 | Login ……………….………………………………………....... | 11 |
| Fig 6-2 | Dashboard …………..…………………………….................. | 13 |
| Fig 6-3 | Admin Complaints …………………………………………… | 13 |
| Fig 6-4 | Admin Information ………….…………………………........ | 13 |
| Fig 6-5 | Admin Canteen Menu……….…………………………........ | 15 |
| Fig 6-6 | Admin Applications….……………………………………… | 16 |
| Fig 6-7 | Student Dashboard…….…………………………………... | 16 |
| Fig 6-8 | Student Complaints…….…………………………………….. | 17 |
| Fig 6-9 | Student Raise Complaints…………………………………. | 18 |
| Fig 6-10 | Student Applications...…………………………………….. | 21 |
| Fig 6-11 | Student Raise Applications……………………………….. | 23 |
| Fig 6-12 | Student Canteen Menu…………………….……………… | 29 |

xi

**CHAPTER 1**

**INTRODUCTION**

**Name of the project:**

HOSTEL MANAGEMENT SYSTEM

**Problem Statement:**

Efficiently managing a hostel involves a multitude of tasks, including student information management, complaint resolution, room allocation, notice management, and canteen menu updates. Traditional manual methods for these tasks can be time-consuming, error-prone, and lack the necessary efficiency. Therefore, there is a crucial need for a comprehensive Hostel Management System (HMS) that can automate and streamline these processes, thereby enhancing overall hostel administration efficiency.

**Objective and scopes of the project:**

The primary objective of the Hostel Management System is to automate and simplify hostel-related tasks. The scope of the project includes:

1. **Student Information Management:** This feature enables the addition and editing of student information. It allows hostel administrators to maintain accurate and up-to-date records of all students residing in the hostel. Information such as personal details, contact information, room allocation, and any other relevant data can be stored and managed through this module.
2. **Complaint Resolution:** This feature provides a platform for students to submit complaints or grievances they may have regarding hostel facilities, services, or other issues. Hostel staff can efficiently address and resolve these complaints by accessing the platform, tracking the status of each complaint, assigning tasks for resolution, and providing updates to the students involved. This enhances communication between students and staff and ensures prompt resolution of issues.

1

1. **Room Allocation:** A room assignment system should efficiently allocate rooms considering factors like academic year, preferences, and any other relevant criteria. Additionally, it should offer scalable infrastructure to seamlessly incorporate new rooms while maintaining a straightforward interface for administrators to manage allocations effectively.
2. **Notice Management:** This feature integrates a system for adding and editing notices to communicate important information to hostel residents. Administrators can create notices regarding upcoming events, policy changes, or any other relevant announcements.
3. **Canteen Menu Management**: This feature allows hostel administrators to add and edit the hostel canteen menu, providing residents with up-to-date information about available food options. The menu can be categorized by meal type (e.g., breakfast, lunch, dinner) and include details such as dishes, prices, special offers, and dietary information. By keeping the menu current and easily accessible, residents can make informed choices about their meals and plan accordingly.

**The system helps the admin to:**

1. **Allot different students to their different hostels:** The system streamlines the process of assigning students to their respective hostels based on various criteria such as gender, academic year, preferences, etc.

Admins can access a module for room allocation where they can view available rooms, assign students to vacant rooms, and manage room occupancy.

This feature ensures efficient utilization of hostel space and facilitates the smooth transition of students into their accommodations.

1. **Vacate the student from the hostels**: Admins have the capability to vacate students from their hostel rooms when necessary. This could be due to the end of the academic term, student withdrawal, or disciplinary reasons.

The system provides a user-friendly interface for admins to select the student(s) to be vacated and update their room status accordingly.

This ensures accurate records of hostel occupancy and helps in preparing rooms for new students or maintenance purposes.

1. **Edit the details of the students and modify the student's records:** Admins have the authority to edit and modify student records as needed to maintain accurate and up-to-date information.

The system provides functionalities for admins to access and update student details such as personal information, contact details, room assignments, and any other relevant data.

2

This capability allows admins to address changes in student circumstances, correct any inaccuracies in records, and ensure the integrity of the student database.

Overall, these functionalities empower admins to efficiently manage hostel operations, maintain accurate records, and provide quality service to students residing in the hostels. By centralizing these tasks within the hostel management system, admins can streamline administrative processes, enhance communication, and improve overall hostel management effectiveness.

**Methodology:**

Here's an explanation of each phase in the software development process for the hostel management system:

1. **Requirements Analysis:** This phase involves gathering and documenting the functional and non-functional requirements for each module. For example:
   1. **Student Information:** Requirements may include the ability to add/edit student details, manage room assignments, and track student history.
   2. **Complaint Resolution:** Requirements may include a complaint submission form, a notification system for staff, and tracking resolution status.
   3. **Room Allocation:** Requirements may include room assignment based on criteria, flexibility for adding new rooms, and a user-friendly interface for administrators.
   4. **Notice Management:** Requirements may include notice creation/editing and targeting specific groups.
   5. **Canteen Menu Management:** Requirements may include menu creation/editing, pricing management, and dietary information.
   6. **Application :** requirement may include application section so student can apply for application.
2. **System Design:** In this phase, the architecture of the system is planned, including the division of modules, communication protocols, and database schema design. User interfaces for each module are also designed to ensure usability and efficiency. For example, the Student Information module may have forms for data entry/editing, while the Complaint Resolution module may have a dashboard for staff to manage complaints.
3. **Implementation:** Development of the system using suitable technologies like ReactJS for front-end, NodeJS for back-end. A relational database like PostgreSQL is chosen to store data securely.

Each module is implemented according to the design specifications, ensuring that it meets the documented requirements.

3

1. **Testing:** Thorough testing is conducted to ensure the functionality, security, and user experience of each module. This includes unit testing, integration testing, security testing, and user acceptance testing.

Testing ensures that the system works as expected, is secure from potential threats, and provides a positive user experience.

1. **Deployment:** Once testing is complete and any issues are addressed, the system is deployed on a web server accessible to authorized users.

Deployment involves setting up the necessary infrastructure, configuring security measures, and ensuring high availability and scalability of the system.

Users are provided with access credentials and any necessary training to use the system effectively.

4

**CHAPTER 2**

**SYSTEM ANALYSIS**

In a System Analysis report for a Hostel Management System project, you would typically outline the analysis process, requirements gathering, and the proposed system's specifications. Here's an explanation of what each section might entail:

1. **Introduction:** Provide an overview of the project, including its purpose, scope, and objectives. Describe the significance of implementing a hostel management system and the benefits it offers to stakeholders such as hostel administrators, staff, and students.
2. **Project Background:** Provide background information on the current hostel management process, highlighting any existing challenges or inefficiencies. Explain the need for a more streamlined and automated system to improve hostel operations and enhance user experience.
3. **System Requirements:** Detail the functional and non-functional requirements gathered through stakeholder interviews, surveys, and analysis of existing systems. Functional requirements should include features such as student information management, room allocation, complaint resolution, fee management, etc. Non-functional requirements should cover aspects such as performance, security, scalability, usability, etc.
4. **Use Case Analysis**: Present use case diagrams and descriptions illustrating the interactions between system actors (e.g., students, hostel staff, administrators) and the proposed system. Identify primary use cases such as student registration, room allocation, complaint submission, fee payment, etc., along with their associated actors and interactions.
5. **Data Flow Diagrams (DFDs):** Develop DFDs to illustrate the flow of data within the system, showing processes, data stores, and data flows between system components. Use level 0 DFD to depict the high-level overview of the system, and subsequent levels to provide more detailed views of specific processes.
6. **Entity-Relationship Diagram (ERD):** Create an ERD to model the relationships between different entities in the system, such as students, rooms, complaints, fees, etc. Define entity attributes and cardinality constraints to accurately represent the data model.

5

1. **System Design Considerations:** Discuss the architectural design decisions, including the choice of technologies, frameworks, and platforms. Describe the proposed system architecture, including the front-end interface (e.g., web-based UI), back-end server, database management system, etc.
2. **Conclusion:** Summarize the key findings of the system analysis phase, emphasizing the identified requirements and design considerations. Provide insights into how the proposed hostel management system addresses the current challenges and meets the needs of stakeholders.
3. **References:** Include any references or sources consulted during the system analysis process, such as academic papers, industry reports, or relevant documentation.

6

**CHAPTER 3**

**REQUIREMENTS ANALYSIS**

1. **Student Information Module:**

**Functional Requirements:**

* 1. **Add/Edit Student Information:** Allow hostel staff to add new students to the system and edit existing student details such as name, contact information, emergency contact, and personal identification details.
  2. **View Student Records:** Provide a feature to view comprehensive student records including personal details, room allocation, fee payment status, and any relevant notes or remarks.
  3. **Room Assignment:** Enable staff to assign rooms to students based on academic year, and room availability.
  4. **Student Search:** Implement a search functionality to quickly locate student records based on various search criteria such as name, ID number, or room number.

**Non-Functional Requirements:**

* 1. **Security:** Ensure that student information is securely stored and accessible only to authorized users with appropriate permissions.
  2. **Scalability:** Design the system to accommodate a large number of students and records without compromising performance.
  3. **Usability:** Develop a user-friendly interface with intuitive navigation and clear data presentation to facilitate easy access and management of student information.

1. **Complaint Resolution Module:**

**Functional Requirements:**

7

1. **Submit Complaints:** Provide students with a platform to submit complaints or grievances regarding hostel facilities, services, or other issues.
2. **Complaint Tracking:** Enable hostel staff to track the status of each complaint, assign tasks for resolution, and provide updates to the students involved.
3. **Priority Setting:** Implement a system for prioritizing complaints based on urgency and severity to ensure timely resolution of critical issues.

**Non-Functional Requirements:**

* 1. **Reliability:** Ensure that the complaint resolution system is reliable and available 24/7 to handle complaints promptly.
  2. **Audit Trail:** Maintain an audit trail of all complaint-related actions including submission, assignment, updates, and resolution for accountability and transparency.
  3. **Accessibility**: Design the complaint submission interface to be accessible to students with disabilities, complying with accessibility standards.

1. **Room Allocation Module:**

**Functional Requirements:**

* 1. **Automated Room Allocation:** Develop algorithms to automate the process of assigning rooms to students based on predefined criteria such as, academic year, and preferences.
  2. **Manual Room Assignment:** Allow hostel staff to manually assign rooms to students as needed, overriding automated allocations if necessary.
  3. **Room Availability:** Display real-time information on room availability, including vacant rooms, occupied rooms, and upcoming vacancies.
  4. **Room Transfer**: Implement a feature for students to request room transfers, with approval and processing by hostel staff.

**Non-Functional Requirements:**

8

* 1. **Efficiency:** Ensure that the room allocation process is efficient and completed in a timely manner to accommodate new students and room change requests.
  2. **Scalability:** Design the system to handle fluctuations in hostel occupancy and accommodate additional rooms or buildings as the hostel expands.
  3. **Fairness:** Ensure fairness and transparency in the room allocation process, avoiding biases or favoritism.

1. **Notice Management Module:**

**Functional Requirements:**

* 1. **Add/Edit Notices:** Enable hostel staff to create, edit, and publish notices regarding upcoming events, maintenance schedules, policy changes, or other important announcements.
  2. **Categorization:** Allow notices to be categorized based on topic or audience (e.g., academic, social, administrative) for better organization and targeted communication.
  3. **Scheduling:** Implement a feature to schedule notices for display during specific times or dates, ensuring timely communication with hostel residents.

**Non-Functional Requirements:**

1. **Reliability**: Ensure that notices are reliably delivered to all residents without fail, with redundancy measures in place to handle system failures or disruptions.
2. **Usability:** Design the notice management interface to be intuitive and easy to use, allowing staff to create and publish notices quickly and efficiently.
3. **Customization:** Provide customization options for notice formatting and design to maintain branding consistency and visual appeal.

9

1. **Canteen Menu Management Module:**

**Functional Requirements:**

* 1. **Add/Edit Menu Items**: Allow hostel staff to add, edit, and remove food items from the canteen menu, including details such as name, description, price, and dietary information.
  2. **Menu Categorization:** Organize menu items into categories such as breakfast, lunch, dinner, snacks, beverages, etc., for easy navigation and selection.
  3. **Menu Availability:** Display real-time information on available menu items, including daily specials, out-of-stock items, and upcoming additions.

**Non-Functional Requirements:**

1. **Accuracy:** Ensure that the menu information is accurate and up-to-date, reflecting changes in availability, prices, and special offers in real-time.
2. **Performance:** Design the menu management system to handle high volumes of menu items and user interactions without experiencing performance degradation.
3. **Integration:** Integrate the menu management module with other systems such as student information and fee payment for seamless data exchange and synchronization.

10

**CHAPTER 4**

**USER PANEL**

**Student Panel:**

1. **Login/Registration:** During registration, students provide essential personal details such as their name, contact information, hostel ID, and any other required information. The system verifies and stores this information securely for future use.

Upon successful registration, students can log in securely using their credentials, typically a username and password.

1. **Home:** The home page serves as the central hub for users once they log in, offering a snapshot of key information and easy navigation to essential features. It often includes personalized content tailored to the user's preferences or recent activity, such as notifications, recent updates, or recommended actions.

Additionally, it provides quick access to commonly used functionalities, such as search, account settings, or frequently visited sections. The design aims for intuitive usability, with clear layout and concise presentation of information to streamline user interaction. Overall, the home page acts as a gateway to the platform's core functionalities, enhancing user experience and facilitating efficient engagement with the system.

1. **Room Booking:** Students can browse through a list of available rooms based on their preferences. They can filter rooms based on criteria like room type (single, double, dormitory), occupancy (number of beds), amenities (AC, attached bathroom, etc.), and location within the hostel.

Detailed information about each room, including rent, amenities, and potential roommates, is provided to help students make informed decisions.

Once a room is selected.

1. **Profile Management:** This feature allows students to maintain and update their personal information stored in the system.

Students can update contact details, emergency contacts, and other relevant information as needed.

Ensures that the hostel administration has accurate and up-to-date information about each student.

1. **Complaints/Suggestions:** Students can use this feature to lodge complaints or provide suggestions related to hostel facilities, maintenance, cleanliness, or any other issues. Provides a direct communication channel between students and the hostel administration, facilitating prompt resolution of concerns and improvement of hostel services.

11

1. **Notice Board:** Important announcements, upcoming events, notices, or any other relevant information from the hostel administration are displayed here.

Helps students stay informed about hostel rules, regulations, deadlines, and events happening within the hostel premises.

1. **Mess Menu:** Displays the daily or weekly menu for meals served in the hostel mess. Allows students to plan their meals in advance and make informed choices.

May include information about special dietary options or any changes in the menu.

**Admin Panel:**

1. **Dashboard:** Provides a comprehensive overview of hostel operations, allowing the admin to monitor occupancy rates, pending tasks, complaints, revenue trends, and other key metrics in real-time.

Enables quick identification of areas needing attention or improvement, facilitating proactive decision-making and efficient management of hostel resources.

1. **Student Management:** Facilitates seamless registration of new students, including capturing essential personal details and preferences for room allocation.

Streamlines the room allocation process by allowing the admin to assign rooms based on factors such as availability, student preferences, and special requirements, ensuring optimal utilization of hostel space.

1. **Room Allocation:** A room assignment system should efficiently allocate rooms considering factors like academic year, preferences, and any other relevant criteria. Additionally, it should offer scalable infrastructure to seamlessly incorporate new rooms while maintaining a straightforward interface for administrators to manage allocations effectively.
2. **Complaints/Suggestions Management:** Provides a platform for students to submit complaints or suggestions regarding hostel facilities, services, or policies, fostering a culture of open communication and continuous improvement.

Allows the admin to view, prioritize, and assign tasks to relevant staff members for resolution, track the status of complaints, and analyze data to identify recurring issues and trends.

12

1. **Notice Management:** Empower the admin to create, edit, and delete notices to be displayed on the student portal, ensuring timely dissemination of important information, announcements, or updates.

Enhances communication between hostel administration and students, promoting transparency and keeping students informed about events, policy changes, or other relevant matters.

1. **Mess Management:** Streamlines the management of hostel mess operations, including menu planning, food inventory tracking, procurement, and quality control, to ensure a consistent and satisfactory dining experience for students.

Enables prompt resolution of any issues related to the hostel mess, such as food quality complaints or supply shortages, thereby maintaining student satisfaction and well-being.

1. **Application:** requirement may include application section so student can apply for application.

The integrated functionalities of the hostel management system streamline operations, fostering efficient communication between students and administration. By offering features like room booking, fee management, and complaint resolution, the system optimizes hostel processes. It enhances transparency, ensures timely responses to student needs, and ultimately elevates the overall management efficacy of the hostel.

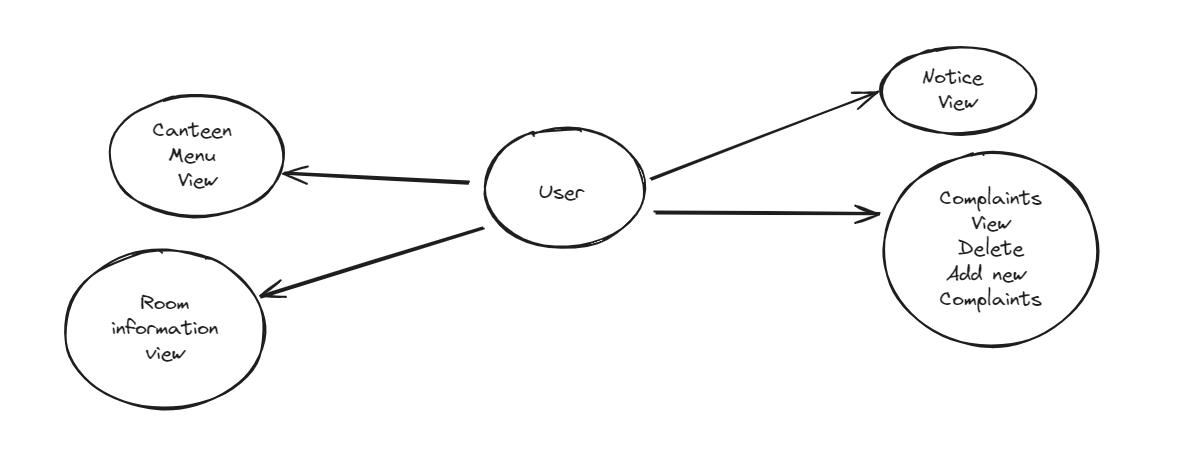
13

**CHAPTER 5**

**DESIGNING**

**User Flow Diagram:**

The image shows a user flow diagram for a student. The student can view the canteen menu, room information, notices, and complaints. They can also add new complaints and delete existing ones.



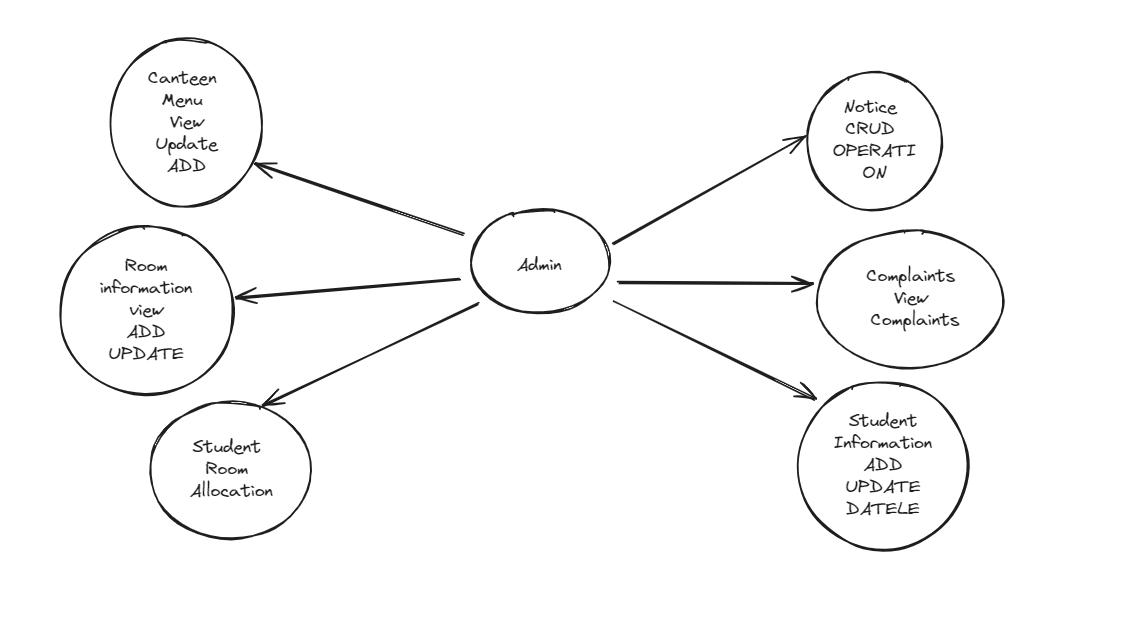
**Fig** 5.1 User Flow diagram

**Admin Flow Diagram:**

The image shows the admin flow of an electronic trading system. It includes the following functionalities:

1. Canteen menu view, update, and add
2. Room information view, add, and update
3. Student room allocation
4. Notice CRUD operation
5. Complaints view and complaints
6. Student information add, update, and delete

14



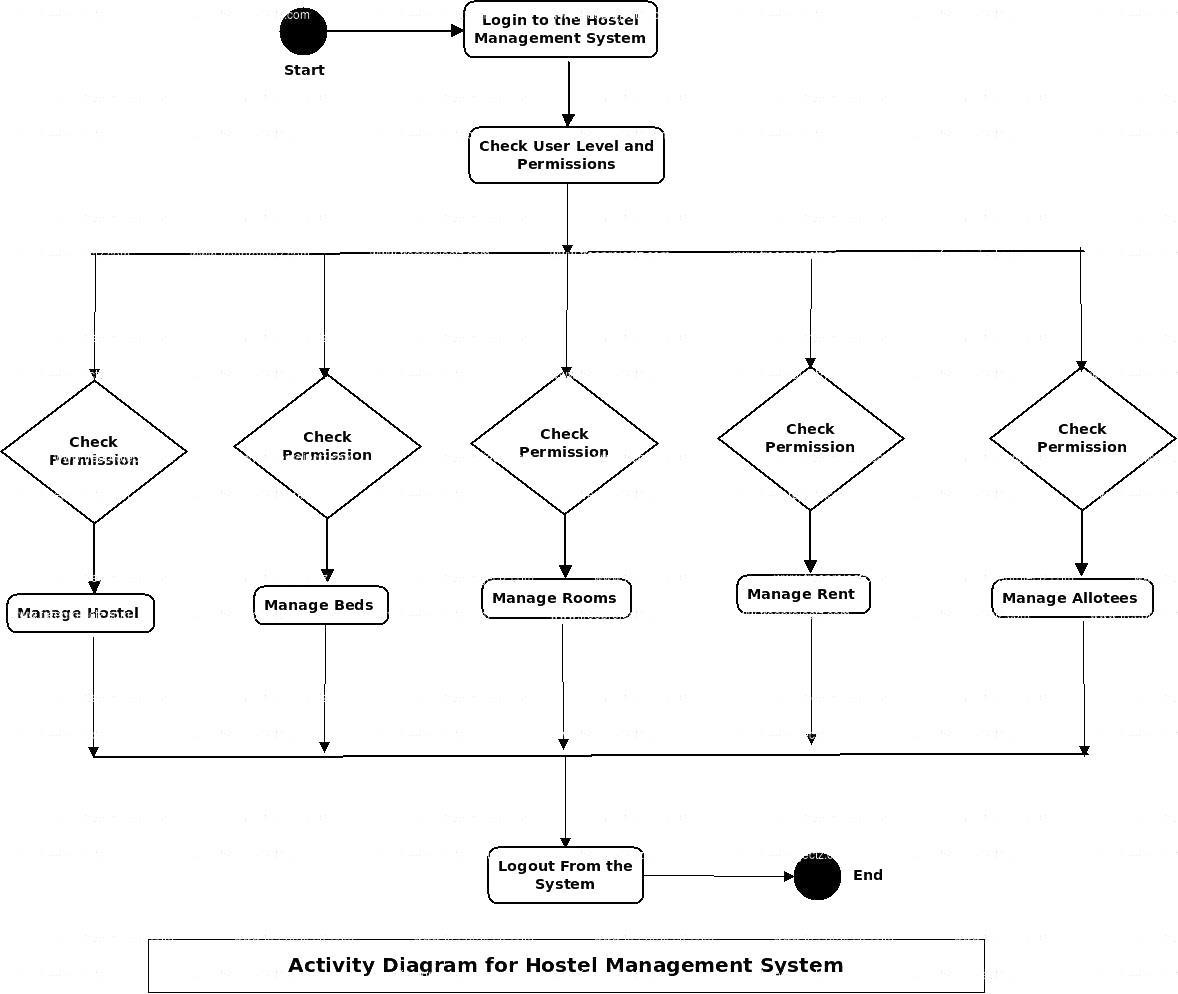
**Fig** 5.2 Admin Flow diagram

**UML Diagram:**

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modem approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we can better understand possible flaws or errors in software or business processes.

15

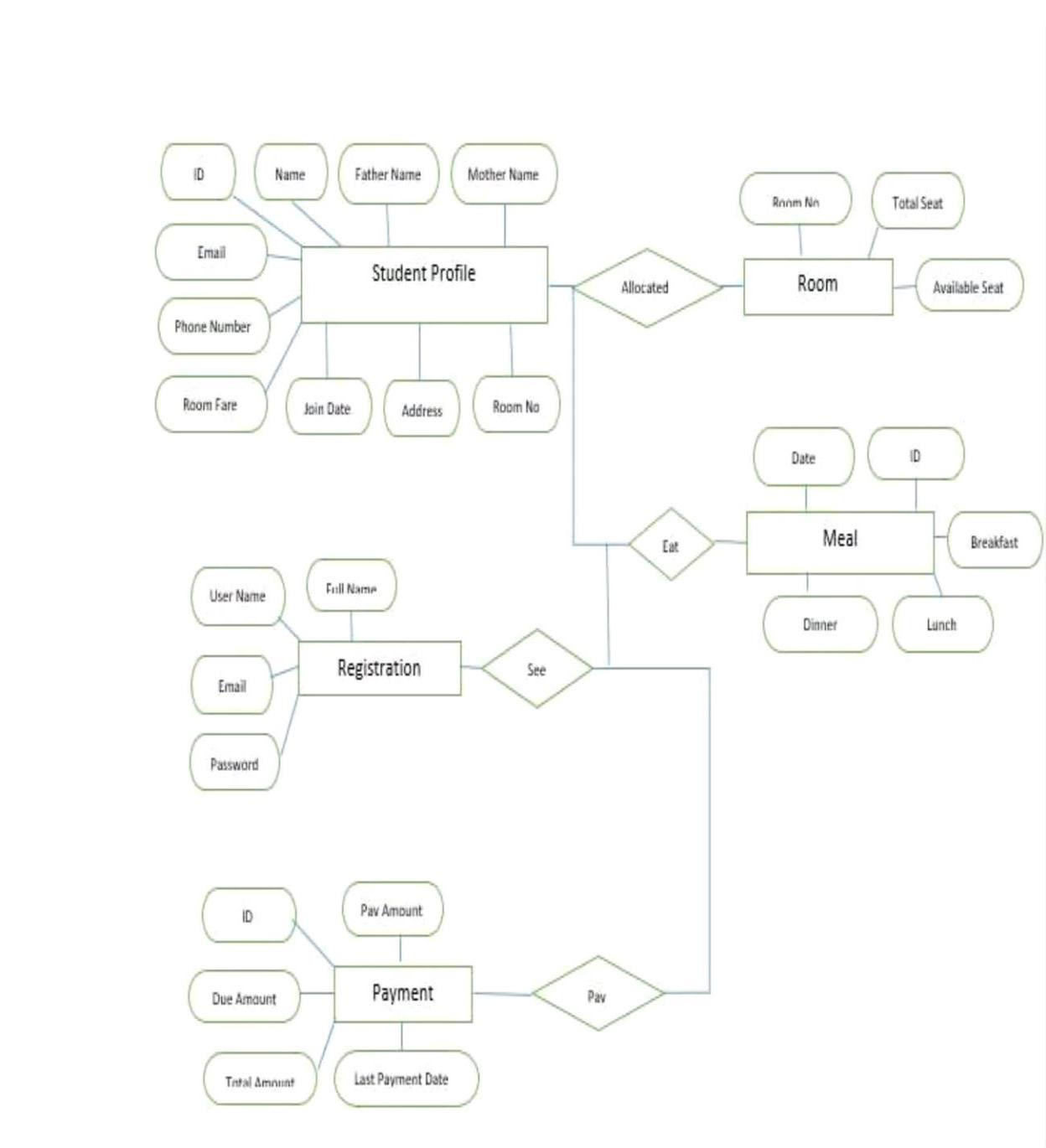


**Fig** 5.3 UML diagram

**ER Diagram:**

The Entity-Relationship (ER) diagram for a hostel management system typically includes entities like Student, Admin, Room, Fee, Complaint, Notice, and perhaps more depending on system complexity. Relationships are depicted between these entities, such as students booking rooms, paying fees, and lodging complaints, which are managed by the admin. The diagram illustrates how data flows within the system, aiding in understanding the structure and interactions of various components. It serves as a visual representation of the database schema, guiding the development and implementation of the hostel management system.

16

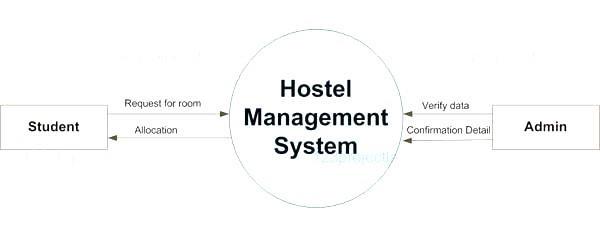


**Fig** 5.4 ER diagram

17

**Data Flow Diagram:**

A data flow diagram (DFD) is a graphical representation of how data flows within a system. It consists of processes, data stores, data flows, and external entities. Processes represent functions or activities that transform data, while data stores are repositories where data is stored. Data flows depict the movement of data between processes, data stores, and external entities. External entities are sources or destinations of data outside the system boundary. DFDs provide a high-level overview of the system's data processing and help identify inputs, outputs, and data transformations. They are useful for understanding system requirements, communication among stakeholders, and designing efficient data handling processes.



**Fig** 5.5 Data Flow diagram 1

**External Entities:** The main external entities are Students and Admins, representing users interacting with the system.

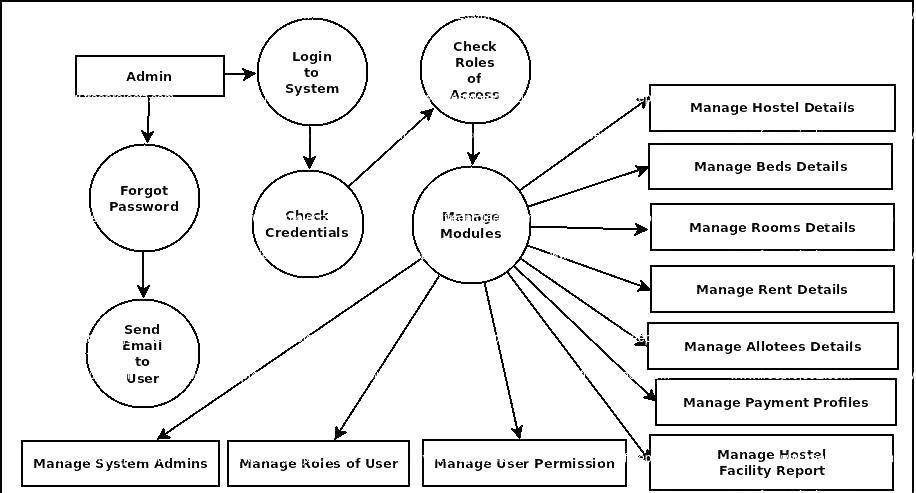
**Processes:** The central process is the Hostel Management System itself, which interacts with the external entities to manage hostel-related activities.

**Data Flows:** Data flows represent the interactions between the external entities and the system, such as student information, room bookings, complaints, and administrative tasks.

**Data Stores:** There may be data stores representing databases or repositories where information is stored, such as student records, room availability, and administrative data.

**System Boundary:** The boundary of the DFD encloses the entire hostel management system, delineating its scope and interactions with external entities.

18



**Fig** 5.6 Data Flow diagram 2

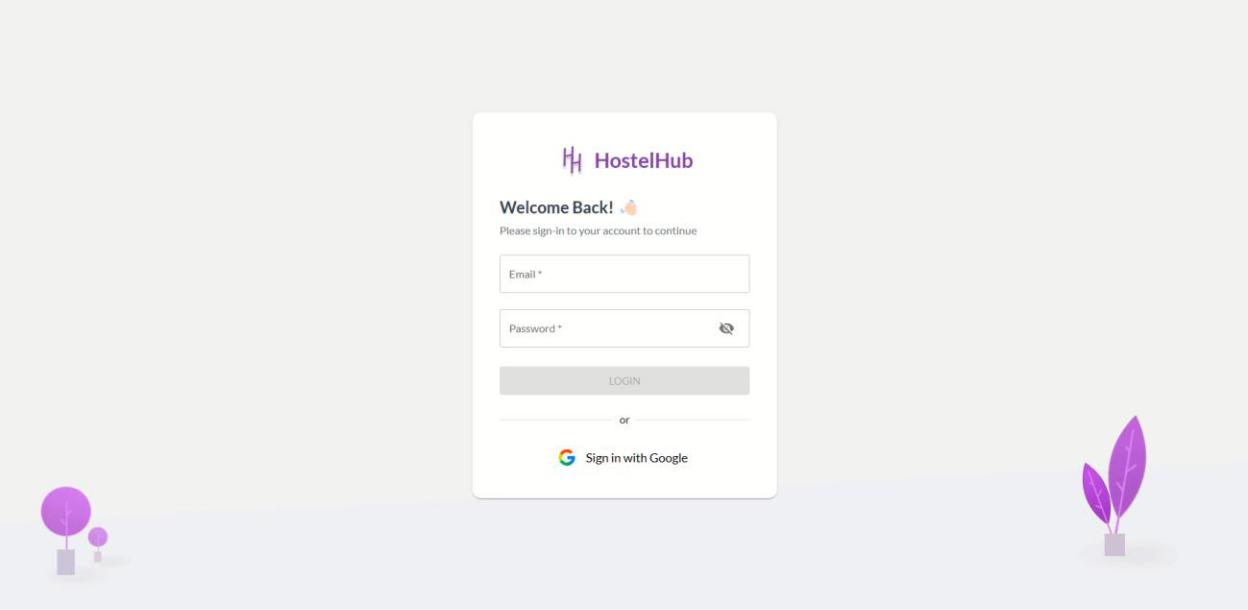
19

**CHAPTER 6**

**USER INTERFACE**

The hostel management system's user interface offers a sleek, intuitive design catering to both students and administrators. With a responsive login page ensuring secure access, it seamlessly transitions users into personalized dashboards. The student interface provides easy room booking, fee payment, and complaint submission, while the admin interface facilitates comprehensive hostel administration, including room allocation, fee management, and notice dissemination, enhancing overall operational efficiency.

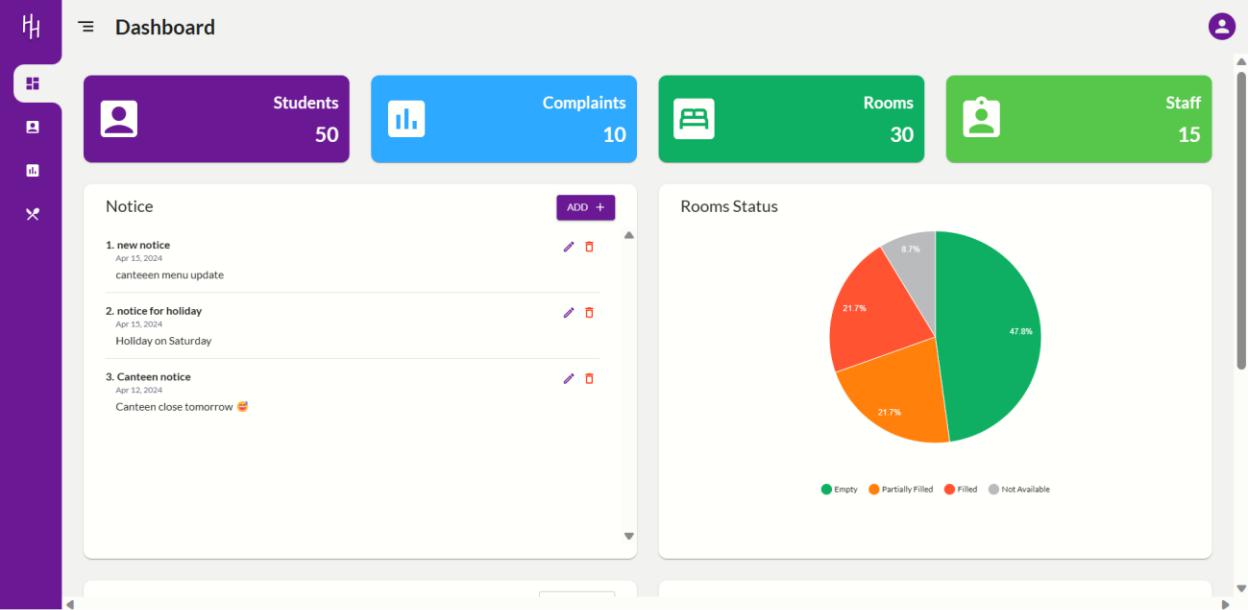
**Login:**



**Fig** 6.1 Login

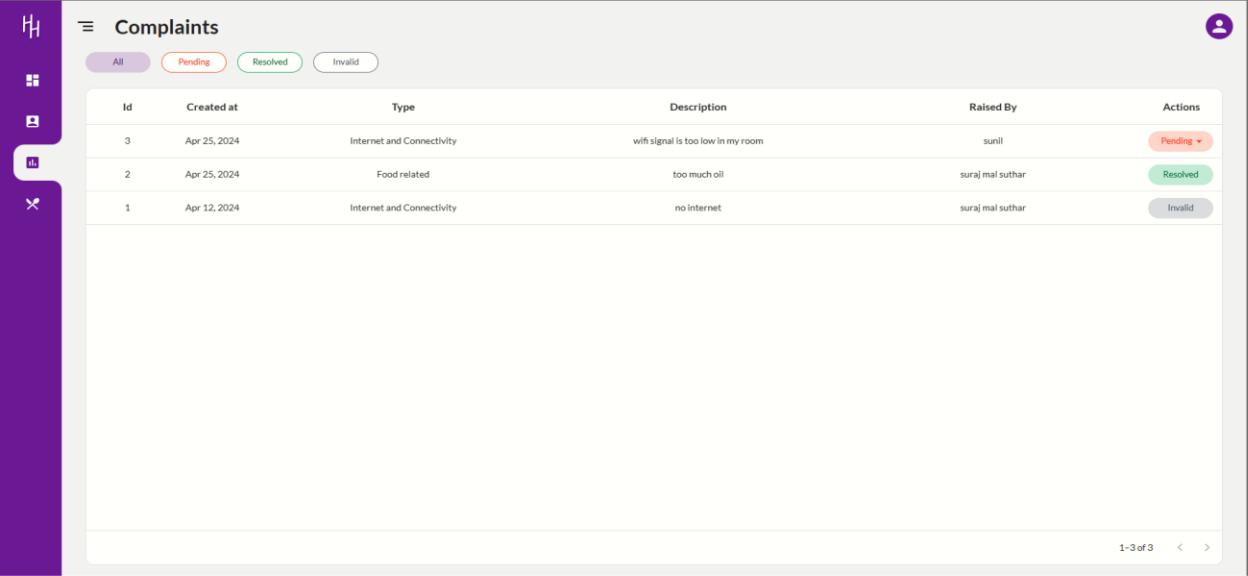
20

**Admin Dashboard:**



**Fig** 6.2 Admin Dashboard

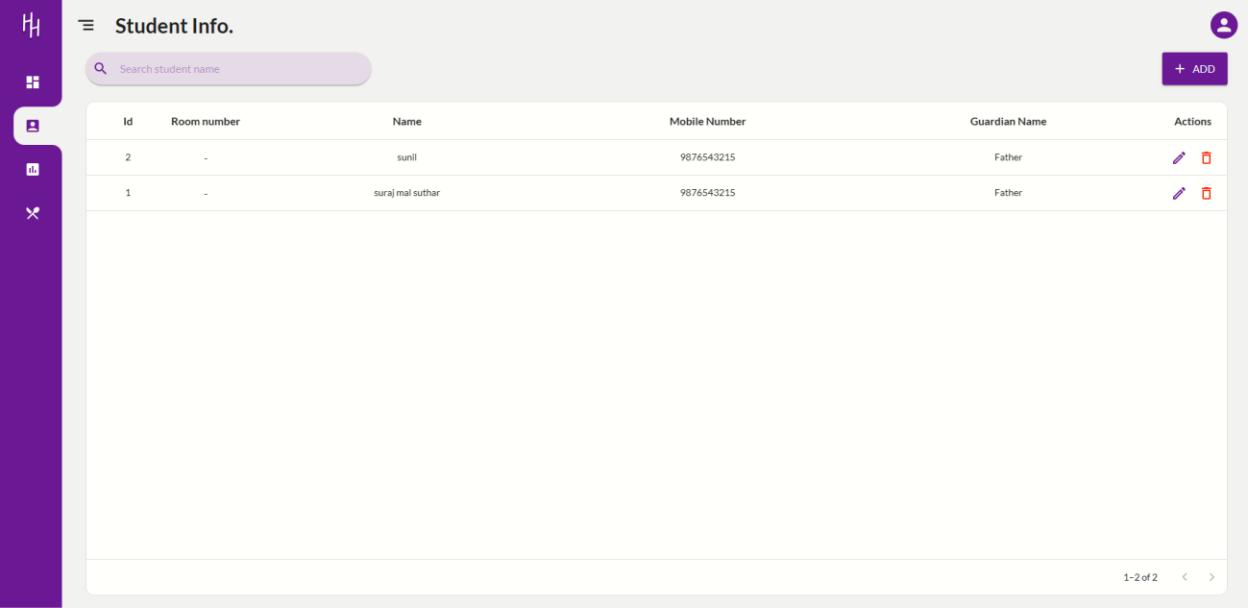
**Admin Complaints:**



**Fig** 6.3 Admin Complaints

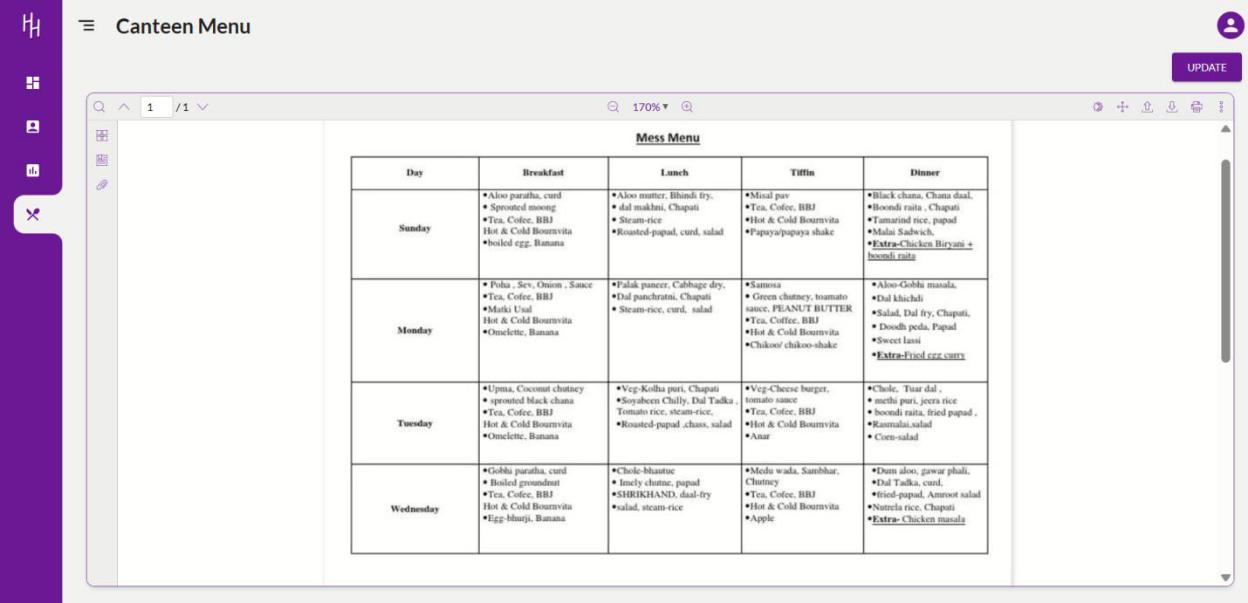
21

**Admin Information:**



**Fig** 6.4 Admin Information

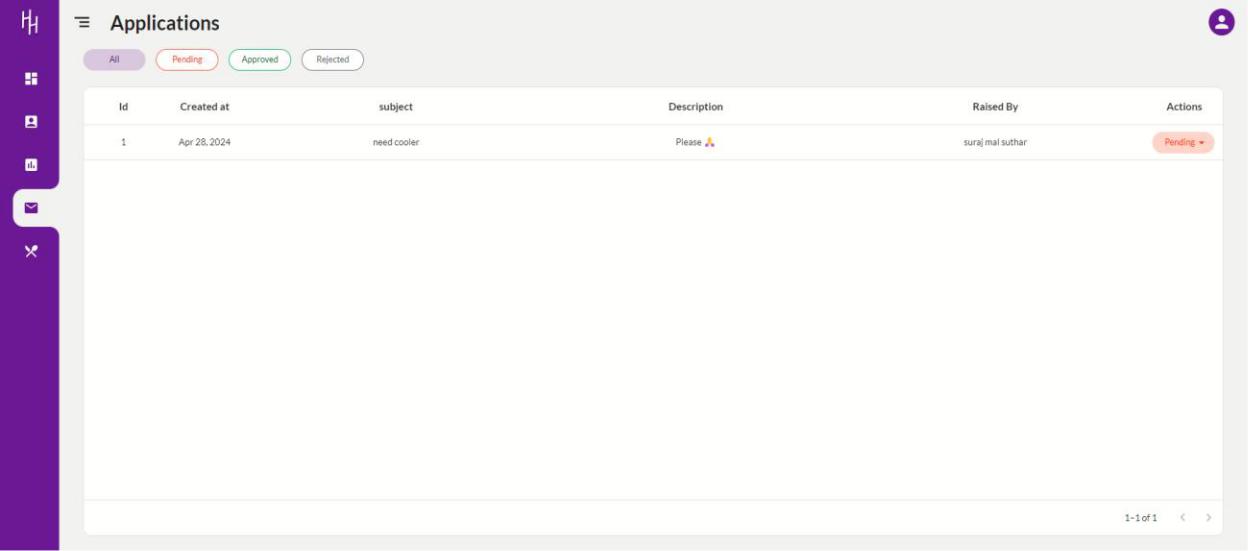
**Admin Canteen Menu:**



**Fig** 6.5 Admin Canteen Menu

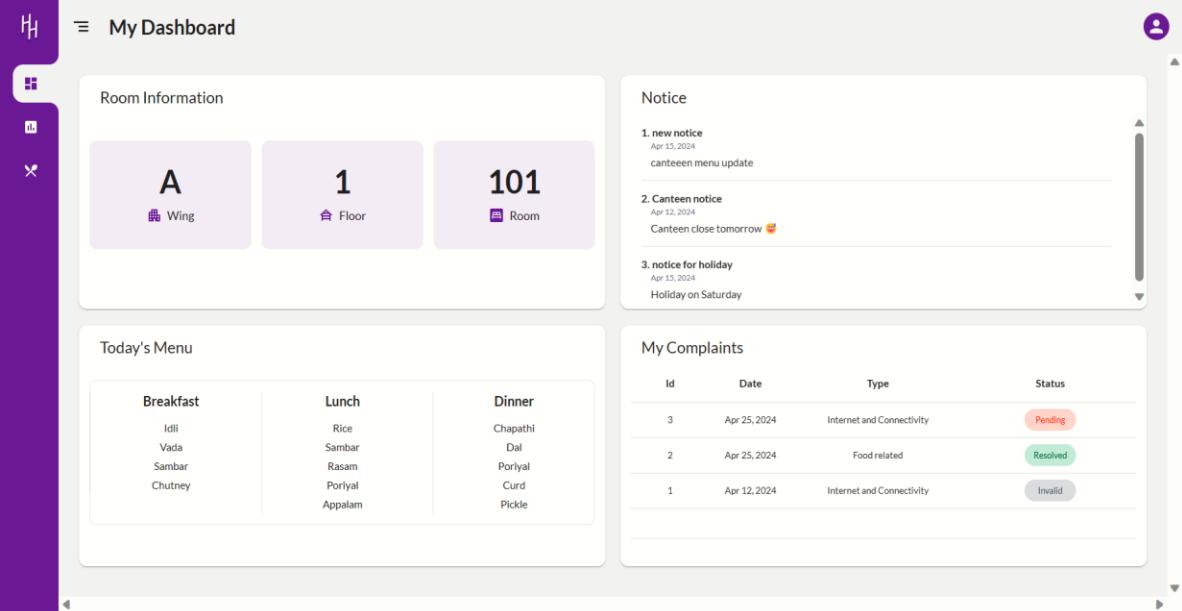
22

**Admin Application:**



**Fig** 6.6 Admin Application

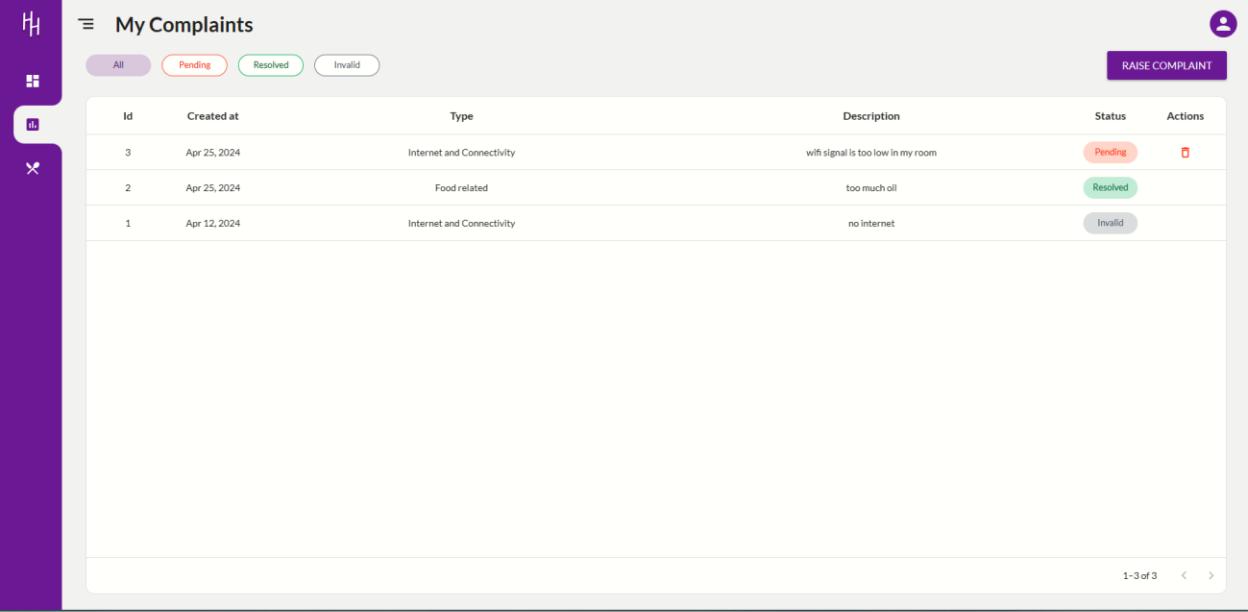
**Student Dashboard:**



**Fig** 6.7 Student Dashboard

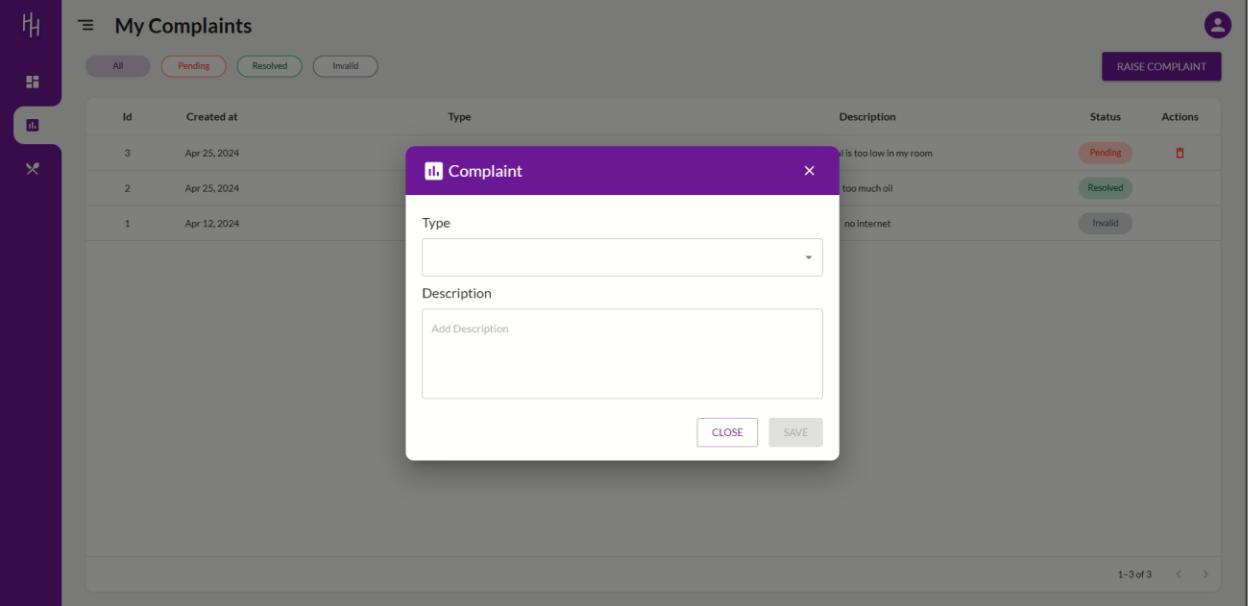
23

**Student Complaints:**



**Fig** 6.8 Student Complaints

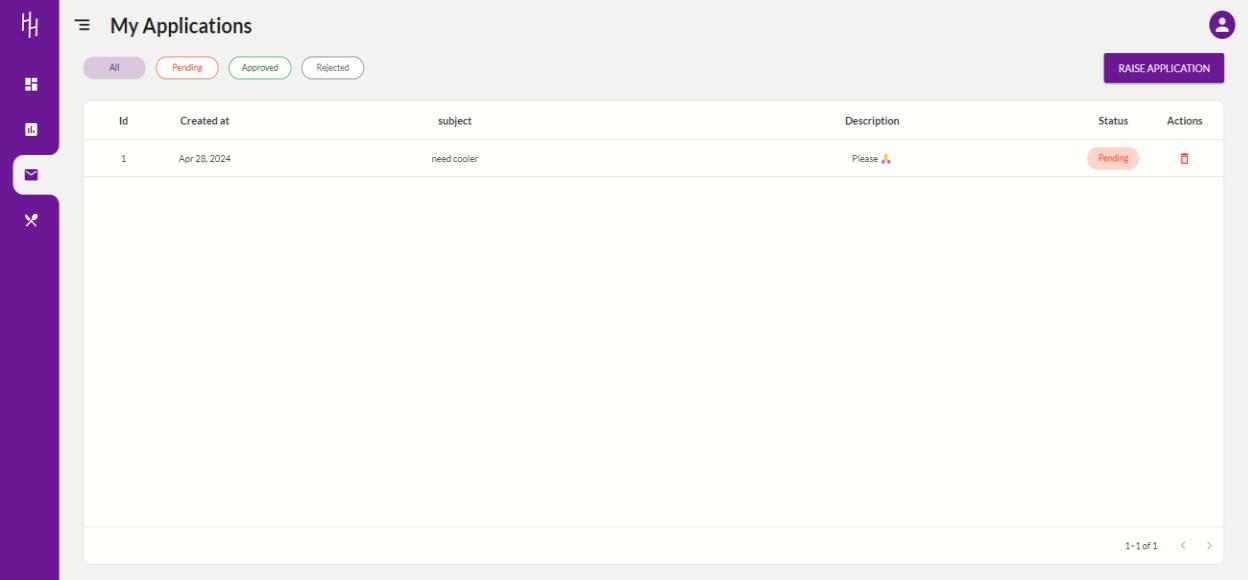
**Student Raise Complaints:**



**Fig** 6.9 Student Raise Complaints

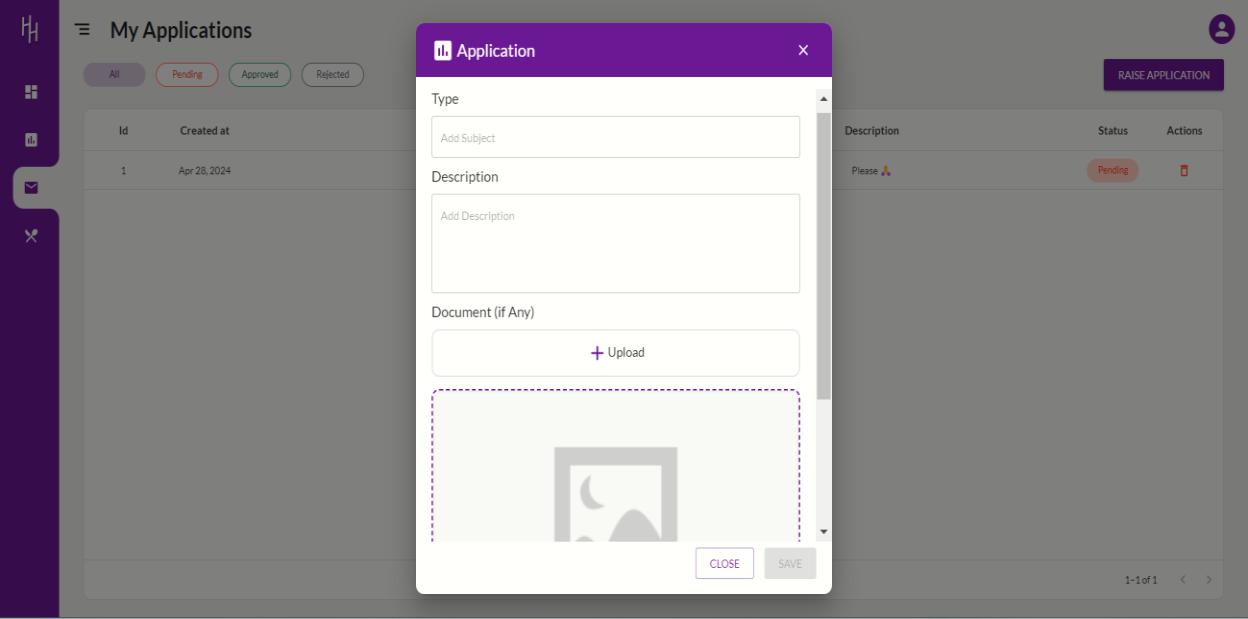
24

**Student Applications:**



**Fig** 6.10 Student Applications

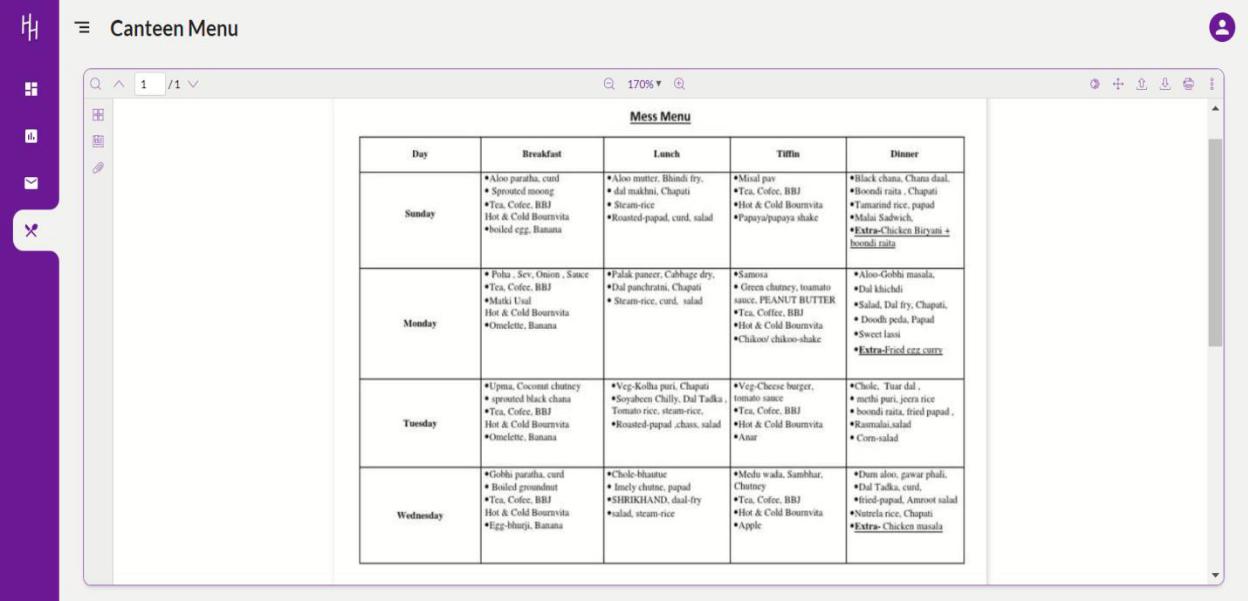
**Student Raise Applications:**



**Fig** 6.11 Student Raise Applications

25

**Student Canteen Menu:**



**Fig** 6.12 Student Canteen Menu

26

**CHAPTER 7**

**LIMITATION AND FUTURE OF THE PROJECT**

**Limitations of the project:**

1. **Scalability:** Hostel management systems may have limitations in terms of scalability, particularly if they are designed for a specific size of hostel or a certain number of residents. As the hostel grows or if there are sudden increases in demand, the system may struggle to accommodate the increased workload efficiently.
2. **Customization:** Pre-built hostel management systems may lack the flexibility to be customized according to the specific needs and workflows of individual hostels. This can be a limitation if a hostel requires certain features or integrations that are not available in the off-the-shelf solution.
3. **Integration:** Hostel management systems may not seamlessly integrate with other software systems used by the hostel or university, such as student information systems or accounting software. This can result in data silos and manual data entry, reducing efficiency and increasing the likelihood of errors.
4. **User Interface:** The user interface of some hostel management systems may not be intuitive or user-friendly, requiring extensive training for staff to effectively use the system. This can lead to resistance to adoption and a slower transition to the new system.
5. **Data Security:** Hostel management systems store sensitive information about students, including personal and financial data. If the system lacks robust security features, it may be vulnerable to data breaches or unauthorized access, potentially exposing residents to privacy risks.
6. **Maintenance and Support:** Like any software system, hostel management systems require regular maintenance and support to ensure they remain functional and up-to-date. However, some systems may have limited support options or rely on a small team of developers, which can lead to delays in addressing issues or implementing new features.
7. **Cost:** Implementing a hostel management system can involve significant upfront costs for purchasing the software, as well as ongoing expenses for maintenance, support, and upgrades. For smaller hostels or those operating on a tight budget, these costs may be prohibitive.

27

1. **Internet Dependency:** Many modern hostel management systems are web-based, which means they require a stable internet connection to function properly. This can be a limitation in areas with unreliable internet access or during network outages.

**Future of the project:**

1. **Integration with RFID or Biometric Systems for Enhanced Security:** RFID (Radio

Frequency Identification): Integrating RFID technology into the hostel management system can provide a more secure and efficient way of managing access to hostel facilities. Each resident can be issued an RFID card or tag that grants them access to specific areas such as their dorm room, common areas, or laundry facilities. This reduces the risk of unauthorized entry and enhances overall security.

Biometric Systems: Biometric authentication methods such as fingerprint or facial recognition can further enhance security by ensuring that only authorized individuals can access hostel facilities. Integrating biometric systems with the hostel management system adds an extra layer of identity verification, reducing the chances of identity theft or fraudulent access.

1. **Integration with Academic Management Systems:** Integrating the hostel management system with academic management systems, such as student information systems (SIS) or learning management systems (LMS), creates a more seamless experience for students and administrators.

This integration allows for automatic syncing of student data, such as enrollment status, course schedules, and academic records, between the hostel management system and the academic management system.

Administrators can use this integrated data to better coordinate room allocations based on students' class schedules, track students' academic performance, and ensure compliance with housing policies and regulations.

1. **Analytics and Reporting Features for Data-Driven Decision-Making:** Implementing analytics and reporting features within the hostel management system enables administrators to gain valuable insights from the data collected by the system.

Administrators can generate reports on various aspects of hostel operations, such as occupancy rates, maintenance requests, billing trends, and resident demographics.

By analyzing these reports, administrators can identify patterns, trends, and areas for improvement, allowing them to make informed decisions to optimize hostel resources, improve resident satisfaction, and enhance overall efficiency.

Predictive analytics can also be utilized to forecast future demand for hostel accommodations, enabling administrators to proactively address capacity issues and plan for future expansion or renovation projects.

28

**Resources of the project:**

1. **Software:**

**ReactJS with TypeScript:** ReactJS is a JavaScript library for building user interfaces, particularly for single-page applications. It allows developers to create reusable UI components.

TypeScript is a superset of JavaScript that adds static typing to the language, which can help catch errors early in the development process and make the code more maintainable.

**Node.js:** Node.js is a JavaScript runtime environment that allows developers to run JavaScript code outside of a web browser. It's commonly used for server-side scripting to build scalable and high-performance web applications.

Node.js uses an event-driven, non-blocking I/O model, making it efficient for handling concurrent connections.

**PostgreSQL:** PostgreSQL is a powerful, open-source relational database management system (RDBMS) known for its reliability, robustness, and support for advanced features such as JSON support, full-text search, and ACID compliance. It's commonly used in web applications for storing and retrieving structured data.

**Strapi:** Strapi is an open-source headless CMS (Content Management System) that allows developers to create, manage, and deploy content APIs quickly. It provides an admin interface for non-technical users to manage content, while also offering a powerful API for developers to fetch and manipulate content programmatically. Strapi can be used to manage various types of content, such as articles, products, images, and more.

1. **Hardware:**

**Web server:** A web server is a computer system that hosts websites or web applications and delivers content to users over the internet. It typically runs web server software such as Apache HTTP Server, Nginx, or Microsoft Internet Information Services (IIS).

The web server hosts the backend code (Node.js in this case) and serves the frontend code (built with ReactJS) to users when they access the application through their web browsers.

29

**Computers with internet access for users:** These are the client devices used by end-users to access the web application.

They could be desktop computers, laptops, tablets, or smartphones with web browsers installed. As long as they have internet access, users can interact with the web application hosted on the web server.

The frontend code (HTML, CSS, JavaScript) is rendered in the users' web browsers, allowing them to view and interact with the application's user interface.

The backend code (Node.js) handles requests from the client, processes data, interacts with the database, and sends responses back to the client.

30

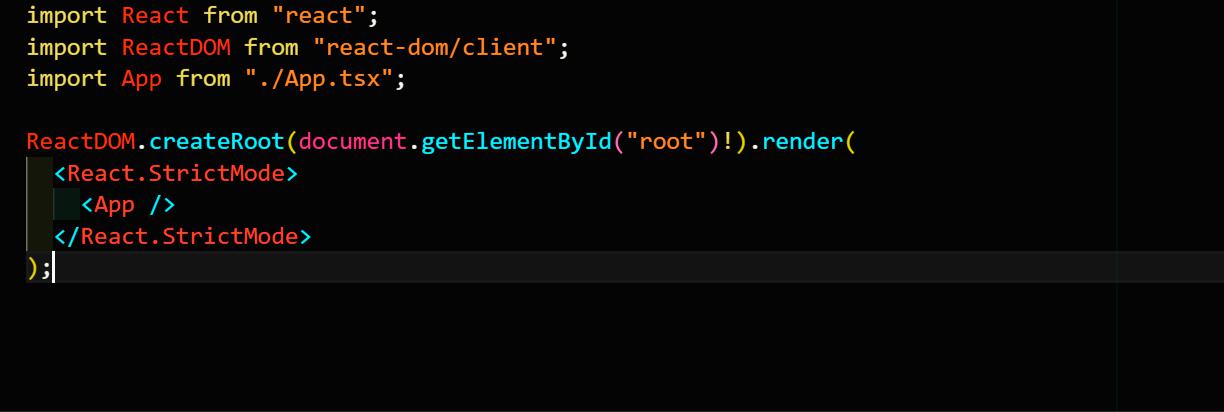
**CONCLUSION**

The Hostel Management System aims to revolutionize hostel administration by providing a user-friendly, automated, and efficient platform. It not only meets the current needs of educational institutions but also provides room for future enhancements and integrations. The project ensures transparency, accuracy, and a seamless experience for both administrators and students in managing hostel affairs. By addressing specific tasks such as student information management, complaint resolution, room allocation, notice management, and canteen menu updates, the system ensures a smoother and more organized hostel experience for both administrators and residents.

31

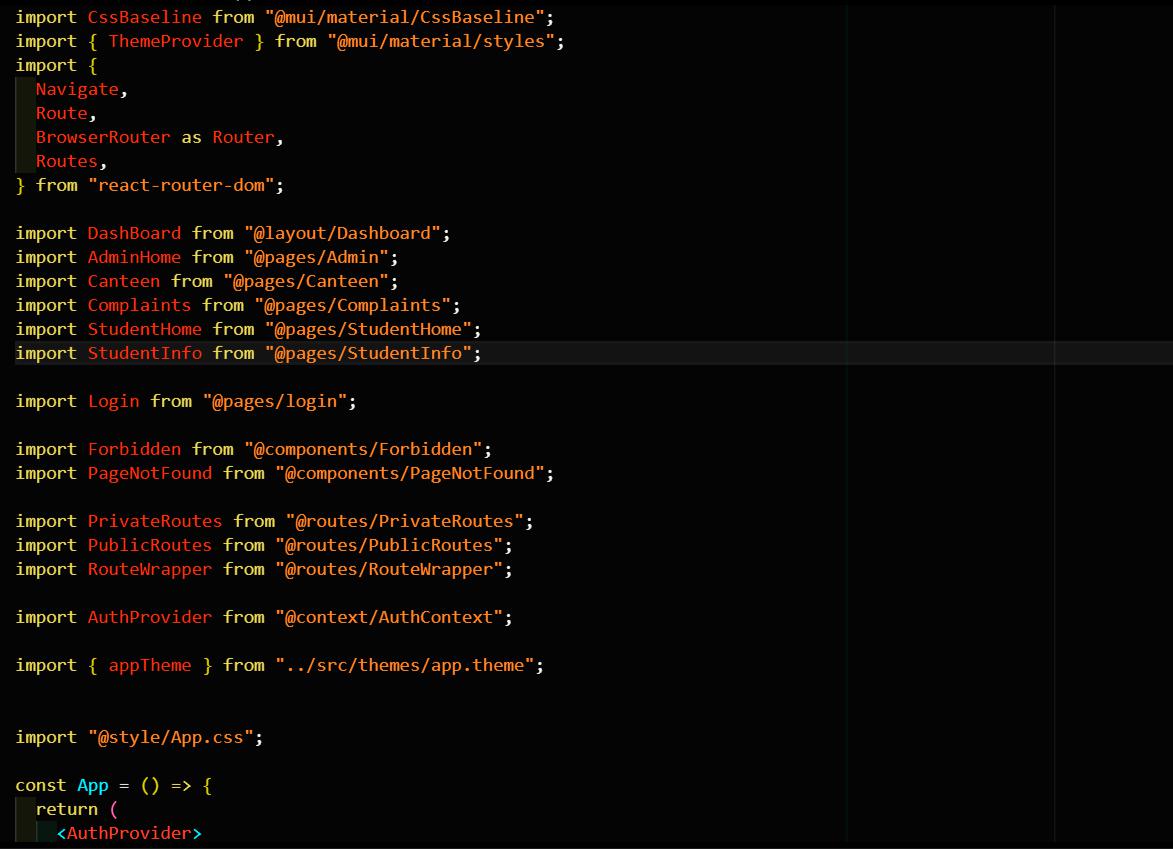
**APPENDICES**

**Appendix 1 – Main.tsx:**

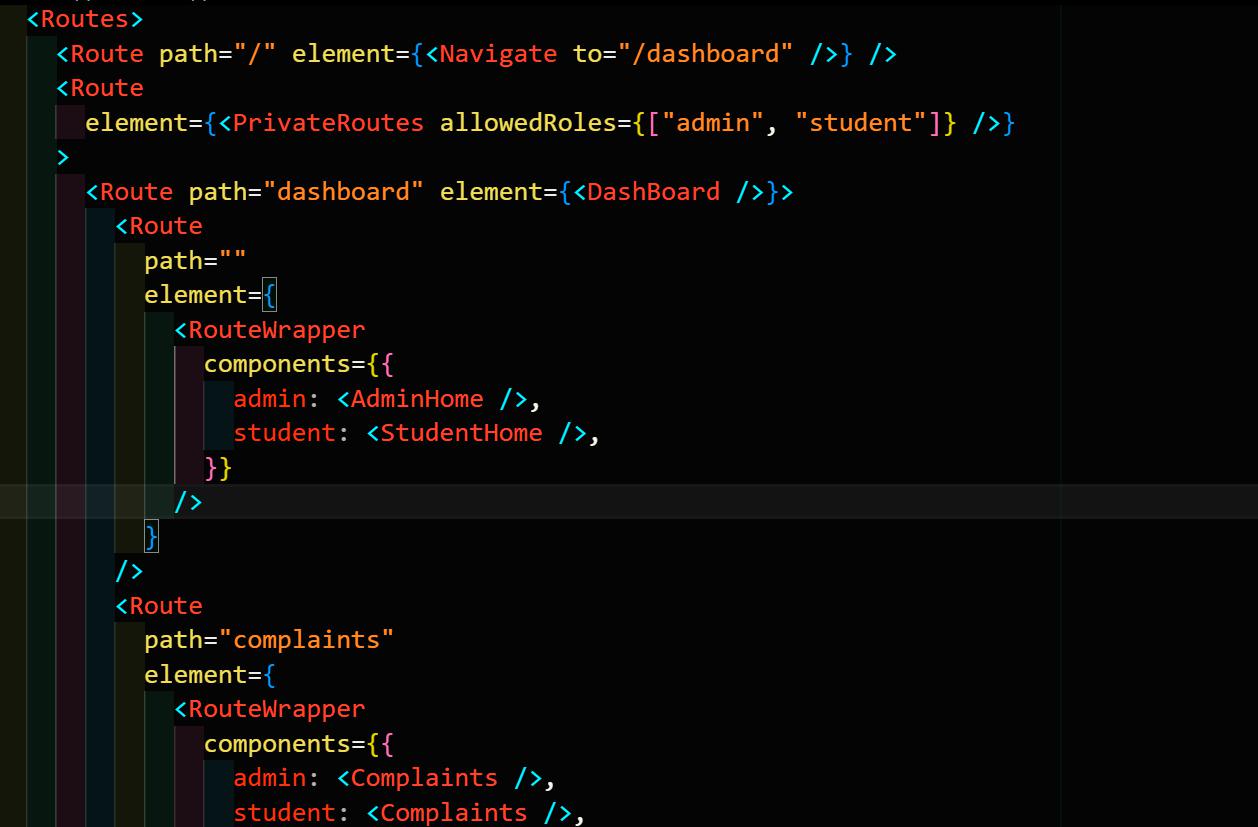


32

**Appendix 2 – App.tsx:**



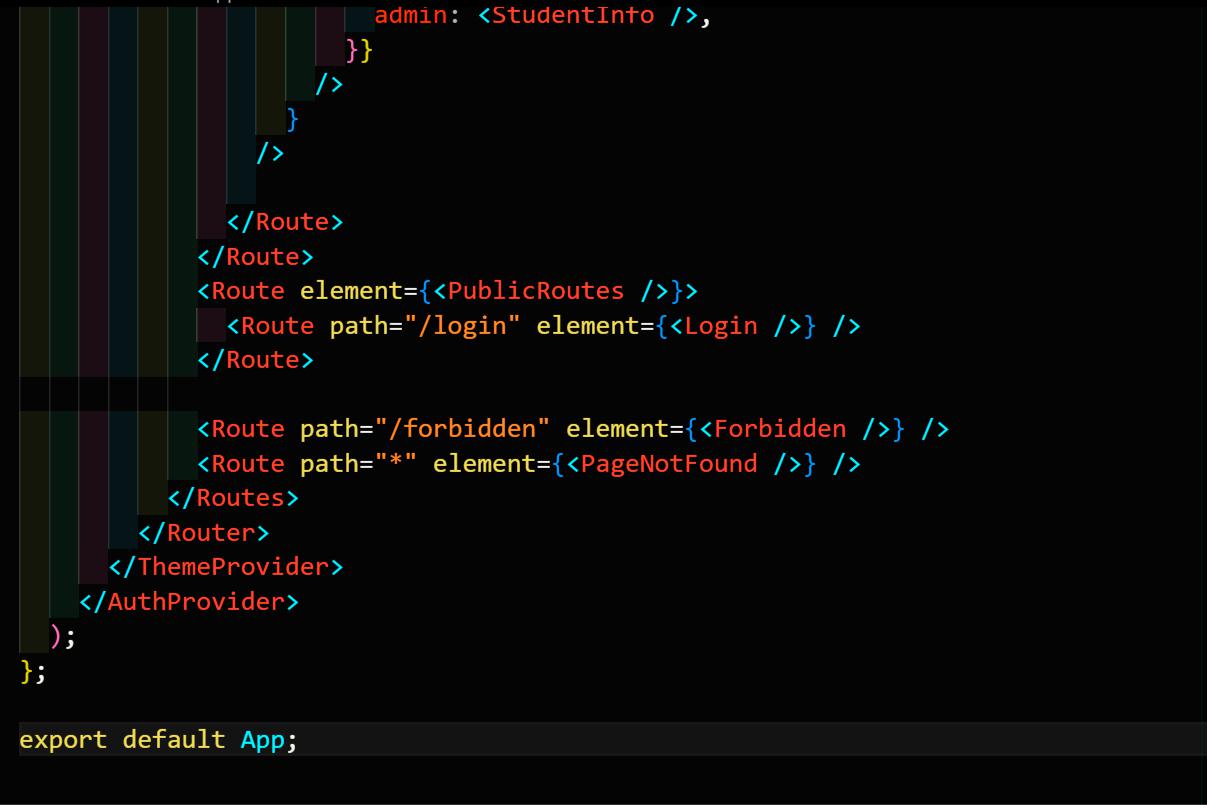
33



34



35



36

**BIBLIOGRAPHY**

**Resource Center :**



<https://strapi.io/resource-center>

**Strapi documentation:**



<https://docs.strapi.io/>

**Strapi tutorials:**



<https://strapi.io/blog/categories/tutorials>

**Strapi blog:**



<https://strapi.io/blog>

**Changelog:**



<https://strapi.io/changelog>

37