

A Field Project Report on

## **FixMyStay: A HOSTEL COMPLAINT MANAGEMENT SYSTEM**

**Submitted**

*In partial fulfillment of the requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE and ENGINEERING**

**By**

R.NAGARAMYA	231FA04488
G.SIVA	231FA04615
BACHALI SAATHVIK	231FA04657
RAHUL KUMAR	231FA04G24

**Under the Guidance of**

**Mr.ch.Amaresh  
Assistant Professor, CSE**



(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF COMPUTING AND INFORMATICS**

**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH**

**(Deemed to be University)**

**Vadlamudi, Guntur -522213, INDIA.**

**April, 2025**



**VIGNAN'S**  
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

### CERTIFICATE

This is to certify that the field project entitled "FixMyStay: A HOSTEL COMPLAINT MANAGEMENT SYSTEM" is being submitted by B. SAATHVIK (231FA04657), G. SIVA (231FA04615), RAHUL KUMAR (231FA04G24) and RAMYA, (231FA04488) in partial fulfilment of the requirements for the degree of Bachelor of Technology (B.Tech.) in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a bonafide work carried out by the aforementioned students under my guidance and supervision.

A handwritten signature in blue ink, appearing to read "Renuka".

Guide

Project Review Committee

A handwritten signature in blue ink, appearing to read "S.P.K".

HOD

of Computer Science & Engineering  
STR Deemed to be University  
VADLAMUDI - 522 213  
Guntur Dist., A.P., India.



**VIGNAN'S**  
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

### DECLARATION

Date: 25/4/25

We hereby declare that the work presented in the field project titled "FixMyStay: A HOSTEL COMPLAINT MANAGEMENT SYSTEM" is the result of our own efforts and investigations.

This project is being submitted under the supervision of **Mr.ch.Amaresh, Assistant Professor, CSE** in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

R.NAGARAMYA	231FA04488	<u>R.Naga Ramya</u>
G.SIVA	231FA04615	<u>G.Siva</u>
BACHALI SAATHVIK	231FA04657	<u>B.Saathvik</u>
RAHUL KUMAR	231FA04G24	<u>Rahul Kumar</u>

## TABLE OF CONTENTS

Chapter No.	Contents	Page No
1	Introduction	6
	1.1 Problem Definition	7
	1.2 Existing System	8
	1.3 Proposed System	8
	1.4 Literature Review	10
2	System Requirements	12
	2.1 Hardware & Software Requirements	13
	2.2 Software Requirements Specification(SRS)	14
3	System Design	17
	3.1 Modules of System	18
	3.2 UML Diagrams	21
4	Implementation	27
	4.1 Sample Code	29
	4.2 Test Cases	33
5	Results	37
	5.1 Output Screens	38
6	Conclusion	42

# 1. INTRODUCTION

The main purpose of this study is to examine the relationship between the use of information technology and the performance of small business.

Information technology has been widely used in various fields, such as business, government, and education.

Small business is one of the most important sectors in the economy, and it is also one of the most affected by the use of information technology.

The use of information technology can help small business to improve their performance, such as increasing efficiency, reducing costs, and improving customer satisfaction.

However, there are also some challenges in the use of information technology for small business, such as the cost of investment, the need for technical skills, and the need for proper management.

In this study, we will examine the relationship between the use of information technology and the performance of small business, and we will also explore the challenges and opportunities in the use of information technology for small business.

We hope that this study will provide useful insights for small business owners and managers to better understand the impact of information technology on their performance.

We also hope that this study will contribute to the development of information technology in small business, and help to improve the overall performance of the sector.

We believe that the use of information technology can bring significant benefits to small business, and we encourage all small business owners and managers to embrace the use of information technology to improve their performance.

We hope that this study will help to promote the use of information technology in small business, and contribute to the development of the sector.

We also hope that this study will help to raise awareness about the importance of information technology in small business, and encourage more small business owners and managers to use it to improve their performance.

We believe that the use of information technology can bring significant benefits to small business, and we encourage all small business owners and managers to embrace the use of information technology to improve their performance.

We hope that this study will help to promote the use of information technology in small business, and contribute to the development of the sector.

We also hope that this study will help to raise awareness about the importance of information technology in small business, and encourage more small business owners and managers to use it to improve their performance.

We believe that the use of information technology can bring significant benefits to small business, and we encourage all small business owners and managers to embrace the use of information technology to improve their performance.

We hope that this study will help to promote the use of information technology in small business, and contribute to the development of the sector.

We also hope that this study will help to raise awareness about the importance of information technology in small business, and encourage more small business owners and managers to use it to improve their performance.

## 1. INTRODUCTION

Hostels serve as a second home for students during their academic journey. Ensuring a clean, safe, and well-maintained hostel environment is crucial for students' comfort, health, and overall well-being. However, managing hostel facilities efficiently is often a challenge, especially when it comes to handling complaints related to electricity, plumbing, internet connectivity, hygiene, or maintenance. In traditional systems, students report complaints by writing them in registers or verbally informing hostel staff. These manual processes are prone to delays, lost entries, miscommunication, and often result in unresolved issues, causing frustration among the residents.

To overcome these inefficiencies, we developed FixMyStay, a web-based Hostel Complaint Management System. This application is designed to simplify and digitize the entire complaint management process. It enables students to submit their complaints anytime from any device with internet access. Students can also monitor the status of their complaints—whether it is pending, in progress, or resolved—thus ensuring transparency and accountability.

FixMyStay not only empowers students with a platform for quick redressal of issues but also helps hostel administrators manage multiple complaints efficiently. Admins can view all complaints in a centralized dashboard, update their statuses, and take prompt action to resolve them. Furthermore, the system includes additional modules such as an Announcements page for hostel updates and a Contacts page to access important staff information, making it a comprehensive communication tool for hostel management.

The aim of this project is to bring digital transformation to hostel administration and create a bridge between students and management through a seamless, user-friendly, and reliable online complaint resolution system.

## 1.1 PROBLEM DEFINITION

In many colleges and universities, especially those with residential facilities, managing student complaints in hostels remains a significant administrative challenge. The current system in most institutions relies on manual processes, where students are expected to either write down their issues in complaint registers or report them verbally to the warden or hostel staff. While this approach may seem straightforward, it is highly inefficient and lacks the structure needed for timely and effective resolution.

Some of the key issues in the existing manual system include:

- **Delayed Complaint Resolution:** Without a proper follow-up mechanism, complaints often go unnoticed or unresolved for extended periods.
- **Loss or Misplacement of Complaints:** Written records can be easily misplaced, ignored, or lost, especially when there is a large volume of entries.
- **Lack of Transparency:** Students have no way of knowing the progress or status of their complaints, leading to frustration and loss of trust in the system.
- **Increased Administrative Workload:** Hostel authorities face difficulty in managing, prioritizing, and responding to multiple complaints simultaneously.

The root of the problem lies in the absence of a structured, digital platform for complaint management. To address this gap, FixMyStay is proposed as a web-based solution that automates and streamlines the entire process. The system aims to improve efficiency by allowing students to submit complaints online, view their status in real-time, and receive timely updates, while also enabling administrators to manage and track complaints systematically.

## 1.2 EXISTING SYSTEM

In many educational institutions, the current system for handling hostel complaints remains largely manual and inefficient. The most commonly used method involves students writing their complaints in physical registers maintained by hostel wardens or administrative staff. While this approach is straightforward, it has numerous drawbacks. Complaints written in registers are often ignored, misplaced, or forgotten, leading to delays in resolution. Additionally, this system lacks any form of tracking or follow-up, leaving students uncertain about whether their issues are being addressed.

Another commonly practiced method is verbal communication, where students directly inform wardens or hostel authorities about their problems. This form of complaint submission is highly unreliable, as it leaves no formal record. It is difficult for hostel staff to recall and manage such complaints effectively, especially when multiple issues are raised simultaneously. There is also a high possibility of miscommunication, and students may feel their concerns are being neglected.

Some institutions have attempted to modernize this process by allowing students to submit complaints via email or instant messaging platforms such as WhatsApp. While these methods offer slightly faster communication, they are not purpose-built for complaint management. These tools lack essential features like complaint categorization, status tracking, user roles, and historical data management. Moreover, such platforms are not secure, and complaints can easily get buried under unrelated messages or forgotten due to the absence of a dedicated tracking system.

Overall, the existing systems fail to deliver the necessary transparency, accountability, and efficiency expected in modern hostel management. Students are left frustrated due to a lack of timely responses, and hostel staff are overburdened without a structured mechanism to prioritize and resolve issues.

## 1.3 PROPOSED MODEL

The proposed solution is FixMyStay, a web-based hostel complaint management system designed to address the inefficiencies of traditional complaint-handling processes in hostels. The system provides an organized, secure, and user-friendly platform that enables students to

file complaints and track their status, while also offering administrators powerful tools to manage complaints, contacts, and announcements effectively.

FixMyStay eliminates the need for physical registers and verbal reporting by digitizing the entire process. It is built using a modern technology stack, with Node.js and Express.js handling the backend, MySQL managing the database, and HTML, CSS, and JavaScript powering the frontend interface. The system is responsive and works seamlessly across devices, ensuring accessibility and convenience for all users.

The model supports two types of users: Students and Admins, each having distinct roles and access to specific functionalities.

Students are the primary users who interact with the system to raise complaints related to their hostel environment. Upon registration and secure login, students are directed to their dashboard where they can submit new complaints by filling in details such as the complaint title, description, category (e.g., maintenance, plumbing, internet), and room number. Students can view a list of all their complaints, including their current status: Pending, In Progress, or Resolved. To maintain data accuracy and allow corrections, students can also edit or delete complaints that are still pending. Apart from complaints, students can access a Contact Page to view updated contact details of hostel authorities such as wardens, caretakers, or electricians. Additionally, students can view important notices or announcements posted by the administrators, helping them stay informed about maintenance schedules or other hostel-related updates.

Admins (typically wardens or authorized hostel staff) use the admin dashboard to manage the complaints submitted by students. From their dashboard, they can view all complaints, filter them by category or status, and change the status of each complaint to reflect its progress. Admins also have the ability to assign specific tasks to maintenance staff, ensuring better resolution and follow-up. The system includes an Announcements Module where admins can post important updates that are instantly visible to all students. This allows for effective mass communication within the hostel. Another critical component for admins is the Contact Management Module, which enables them to add, edit, or delete contact details of key personnel. This ensures that all contact information available to students is accurate and up to date.

The proposed model supports real-time updates, so students receive feedback when the status of their complaint changes or when new announcements are posted. All data—complaints, user details, contacts, and announcements—is securely stored in a MySQL database, which ensures data consistency and integrity.

Key advantages of the proposed model include:

- Faster complaint submission and resolution with structured workflows.
- Transparency in tracking complaint status and administrator actions.
- Improved communication through announcement postings and accessible contact information.
- Centralized management of all hostel-related issues in one digital platform.
- Scalability for future upgrades like mobile applications, SMS/email notifications, or feedback systems.

In summary, FixMyStay transforms the traditional hostel complaint process into a smart, transparent, and responsive system that benefits both students and administrators. It significantly reduces delays, ensures accountability, and creates a more organized hostel environment.

#### 1.4 LITERATURE REVIEW

The traditional process of managing hostel complaints in many educational institutions has long relied on manual systems such as handwritten registers or informal verbal reporting. Several studies and institutional surveys have pointed out the limitations of these methods. Manual systems are prone to data loss, delays in addressing complaints, and a lack of documentation. Without a structured tracking system, students remain unaware of the progress or resolution of their issues, which leads to dissatisfaction and a lack of trust in the administrative process.

Research in the field of service management and digital transformation highlights the benefits of moving from manual to automated systems. For instance, web-based complaint

management platforms are already being widely used in sectors such as municipal governance, corporate customer support, and healthcare services. These platforms have proven effective in reducing complaint resolution time, improving communication, and ensuring accountability through proper logging and tracking mechanisms.

Studies on digital grievance redressal systems emphasize that transparency and accessibility are key to improving user experience. When users are informed about their complaint status and administrators are given tools to track and manage issues, the overall system becomes more efficient and trustworthy. These findings are directly applicable to hostel environments, where students need a reliable way to report and monitor complaints related to their living conditions.

In educational institutions, much of the focus in digital management has historically been on academic performance, room allocation, fee payment, and attendance tracking. While these features are important, complaint management often remains underdeveloped or entirely overlooked. Few systems integrate modules specifically for managing non-academic issues faced by hostel residents. The lack of such systems negatively impacts student well-being and administrative effectiveness.

FixMyStay was conceptualized in response to these gaps. It incorporates findings from multiple studies and best practices from other sectors to build a dedicated, student-centric complaint management system. The platform allows students to file complaints from anywhere, track real-time status updates, and access contact information for hostel authorities. It also equips administrators with tools to manage, update, and resolve complaints efficiently, post announcements, and maintain contact records.

The system aligns with the latest research recommendations for transparent, efficient, and user-friendly digital platforms. By incorporating both a complaint module and a contact management module, FixMyStay stands out as a comprehensive solution for hostel administration, significantly improving the quality of hostel life through accountability, communication, and structured resolution workflows.

## **2.SYSTEM REQUIREMENTS**

## 2.SYSTEM REQUIREMENTS

### 2.1 HARDWARE AND SOFTWARE REQUIREMENTS

The FixMyStay - Hostel Complaint Management System is a web-based application that can be accessed through a browser. However, it requires certain hardware and software configurations for smooth operation, both for end-users (students/admins) and for the server-side environment. These requirements ensure the application runs efficiently without any interruptions.

#### Hardware Requirements

For Client-Side (Students and Admins):

The hardware requirements for users accessing the system through a web browser are minimal. The system is lightweight and can run on most standard computers.

#### *Component Minimum Requirement*

<i>Processor</i>	Intel Core i3 (or equivalent AMD processor)
<i>RAM</i>	4 GB (minimum)
<i>Hard Disk Space</i>	100 GB (for temporary storage and files)
<i>Monitor</i>	15-inch or larger display for better readability
<i>Keyboard &amp; Mouse</i>	Standard input devices
<i>Internet</i>	Minimum 1 Mbps broadband connection for smooth interaction

For Server-Side (Hosting the Backend and Database):

In case the system is deployed on a local or cloud server, the following hardware is recommended for the server machine:

#### *Component Minimum Requirement*

<i>Processor</i>	Intel Core i5 or higher
<i>RAM</i>	8 GB or more
<i>Hard Disk Space</i>	250 GB SSD (for faster data processing and storage)
<i>Network</i>	Stable and fast internet connection (10 Mbps or more)

<b>Backup Devices</b>	External hard drives or cloud backup for database security
-----------------------	--

## Software Requirements

The FixMyStay system depends on a combination of frontend, backend, and database technologies. The following software components are required to develop, deploy, and maintain the system.

### Software Requirement

<i>Operating System</i>	Windows 10 / Linux (Ubuntu) / macOS
<i>Web Browser</i>	Google Chrome / Mozilla Firefox (latest versions)
<i>Frontend Technologies</i>	HTML5, CSS3, JavaScript
<i>Backend Technologies</i>	Node.js (version 14 or higher), Express.js
<i>Database</i>	MySQL (version 5.7 or higher)
<i>Code Editor</i>	Visual Studio Code (recommended)
<i>Version Control (Optional)</i>	Git (for source code management)

### Additional Tools:

Bootstrap (CSS Framework): To make the user interface responsive and mobile-friendly.

Postman: For testing backend API endpoints during development.

The above hardware and software configuration ensure that the FixMyStay system functions smoothly, with optimal speed and performance for both students and hostel staff.

## 2.2 SOFTWARE REQUIREMENTS SPECIFICATIONS

The Software Requirements Specification (SRS) defines the complete functionality, features, and constraints of the FixMyStay - Hostel Complaint Management System. It ensures that the system meets user needs by providing an automated, efficient, and transparent platform for hostel complaint management.

## **Functional Requirements**

### User Registration and Login

- Students register by entering their name, registration number, room number, email, and password.
- Admins have predefined accounts or are created by super admins.
- Secure login with validation is required for both roles.

### Complaint Filing

- Students can submit complaints by entering a title, description, category (Maintenance, Internet, Food, etc.), and room number.
- The system validates complaint forms before submission.

### Complaint Status Tracking

- Students can view the real-time status of each complaint (Pending, In Progress, Resolved).
- Admins update statuses based on progress.

### Admin Panel

- Admins can view, update, and manage all complaints.
- Complaints can be assigned to specific staff or departments.

### Contact Management System

- Admins can add and manage contact details of hostel authorities.
- Students can view contact details for direct communication.

### Edit/Delete Complaint (Student)

- Students can edit or delete complaints if they are still pending.

### Logout

- Both students and admins can log out securely after their session.

---

## **Non-Functional Requirements**

### ➤ Usability

The system must be simple and user-friendly for students and staff.

The UI should have clear navigation, easy-to-fill forms, and readable text.

➤ Reliability

The system should function without crashes or data loss.

All complaints should be accurately stored and retrieved.

➤ Performance

The system should support multiple users filing/viewing complaints simultaneously without lag.

➤ Security

Passwords must be encrypted for security.

Unauthorized users should be restricted from accessing complaints.

Role-based access ensures privacy and data protection.

➤ Scalability

The system should support more users or hostels in the future.

➤ Maintainability

The code should be easy to update for bug fixes and feature enhancements.

---

## User Interface Requirements

- Login/Signup Pages: Simple, validated forms for authentication.
- Student Dashboard: Displays a summary of complaint history with status updates.
- Admin Dashboard: Lists all complaints with status update and assignment options.
- Complaint Form: Well-structured, easy-to-fill form with dropdown menus for category selection.
- Contact Management Page: Displays important hostel contacts for students.

## 3. SYSTEM DESIGN

### 3.1. System Architecture

- The system architecture is divided into three main components:
  - Database layer: Stores all the data required for the system.
  - Application layer: Handles user requests and interacts with the database.
  - User interface layer: Provides a graphical or textual interface for users to interact with the system.

\* Application layer: Handles user requests and interacts with the database.

### 3.2. Database Design

- The database design consists of two main tables:
  - User table: Stores information about registered users, including their ID, name, email, password, and other personal details.
  - Transaction table: Stores information about transactions, including the transaction ID, user ID, amount, date, and time.

\* User table: Stores information about registered users.

\* Transaction table: Stores information about transactions.

\* Primary key: User ID and Transaction ID are primary keys.

\* Data types: User table: ID (int), Name (text), Email (text), Password (text). Transaction table: ID (int), User ID (int), Amount (float), Date (date), Time (time).

### 3.3. Application Layer

## 3.SYSTEM DESIGN

### 3.4. User Interface

- The user interface is designed to be user-friendly and intuitive.
  - It features a clean, modern design with a light blue color scheme.
  - The interface includes a header with the system logo and navigation links.
  - The main area displays a dashboard with real-time data and charts.
  - There are separate sections for user management, transaction history, and reporting.

\* Header: Includes the system logo and navigation links.

\* Main Area: Displays a dashboard with real-time data and charts.

\* User Management: Allows users to manage their account settings and view transaction history.

\* Transaction History: Allows users to view a detailed history of their transactions.

\* Reporting: Allows users to generate reports based on specific criteria.

\* Footer: Includes links to help and support resources.

\* Overall: The user interface is designed to be user-friendly and intuitive.

\* Colors: The user interface uses a light blue color scheme throughout.

\* Fonts: The user interface uses a sans-serif font for all text elements.

\* Buttons: Buttons are clearly labeled and have a distinct visual style.

\* Navigation: Navigation is intuitive, with clear links to different sections of the application.

\* Overall: The user interface is designed to be user-friendly and intuitive.

## 3.SYSTEM DESIGN

The system design of FixMyStay describes the blueprint of the application, explaining how different components interact and how data flows throughout the system. This design phase is crucial as it ensures that the final implementation meets all user and system requirements efficiently and is scalable, maintainable, and user-friendly.

FixMyStay is built on a modular architecture, meaning the system is broken down into several components or modules, each responsible for a specific task. This modular design promotes clarity, maintainability, and scalability, allowing future enhancements and debugging to be done efficiently.

This section presents the core modules of the system and provides a visual representation of the system's architecture through UML diagrams, including the Use Case Diagram, Class Diagram, and Activity Diagram. These diagrams help in understanding the structure, behavior, and interaction between users and system components.

### 3.1 MODULES OF SYSTEM

The FixMyStay system is organized into five key modules, each playing a vital role in ensuring smooth and transparent complaint resolution. Below is a detailed description of each module:

#### 1. User Authentication Module

- This module is responsible for managing the login and registration processes of users.
- Students register by providing personal information such as name, registration number, room number, email, and password.
- Admins have predefined login credentials, often managed by a higher-level administrator or IT support.
- All login attempts are securely validated, and invalid credentials are rejected.
- Upon successful authentication, users are redirected to their respective dashboards based on their roles.

Security is a key concern in this module. Passwords are encrypted using hashing techniques, and access control mechanisms are in place to prevent unauthorized access.

---

## 2. Student Dashboard Module

This module serves as the main interface for students after logging in.

- Students can view all previously filed complaints along with their current statuses: Pending, In Progress, or Resolved.
- They can file new complaints using a structured form that includes fields like title, description, category, and room number.
- Complaints that are still in the "Pending" status can be edited or deleted.
- Students can also view important hostel contacts for further assistance and check announcements posted by the admin.
- A well-designed UI ensures ease of navigation, with responsive design elements for desktop and mobile use.

The dashboard ensures that students stay informed and engaged in the resolution process of their complaints.

---

## 3. Complaint Filing Module

This module handles the actual submission of complaints by students.

- A detailed complaint form captures the essential details.
- Form fields include: title of complaint, description, complaint category (like Water, Electricity, Internet), and room number.
- The form undergoes validation checks to ensure data integrity before being submitted.
- Once submitted, the complaint is stored in the database and marked as "Pending".
- An acknowledgment or confirmation is shown to the student upon successful submission.

This module is crucial as it forms the entry point for the complaint lifecycle in the system.

---

## 4. Admin Dashboard Module

This module is designed for hostel staff or administrators and offers full access to manage the complaint system.

- Admins can view all complaints submitted by students.
- Complaints can be sorted or filtered based on status, category, or date.
- Admins can update the status of a complaint as it progresses through different stages: Pending → In Progress → Resolved.
- Tasks can be assigned to specific staff members, and remarks or comments can be added for transparency.
- The dashboard also allows posting announcements, which are instantly visible to all students.
- Through the Contact Management System, admins can add, update, or delete hostel contacts, ensuring students have access to up-to-date information.

This module empowers the admin with the tools needed to keep the hostel management process streamlined and responsive.

---

## 5. Database Management Module

- This module manages the backend storage and retrieval of all data in the system.
- All user details, complaint records, announcements, and contact details are securely stored in a MySQL database.
- It supports CRUD operations (Create, Read, Update, Delete) efficiently.
- Data security and consistency are ensured through primary and foreign key constraints.
- Backup strategies can be implemented to prevent data loss in case of failure.
- This module also ensures fast querying for smooth performance even under heavy user load.

The database module plays a foundational role in maintaining the reliability and responsiveness of the entire system.

## 3.2 UML DESIGN (UNIFIED MODELLING LANGUAGE)

To better understand the working of FixMyStay, three essential UML diagrams are presented:

Use Case Diagram, Class Diagram, and Activity Diagram. These diagrams visualize the functionality, structure, and processes of the system.

---

### 3.2.1 Use Case Diagram

The Use Case Diagram illustrates the interactions between users (students and admins) and the system.

#### Actors:

- Student
- Admin

#### Use Cases for Students:

- Login
- Register
- File Complaint
- View Complaint Status
- View Contacts
- View Announcements
- Logout

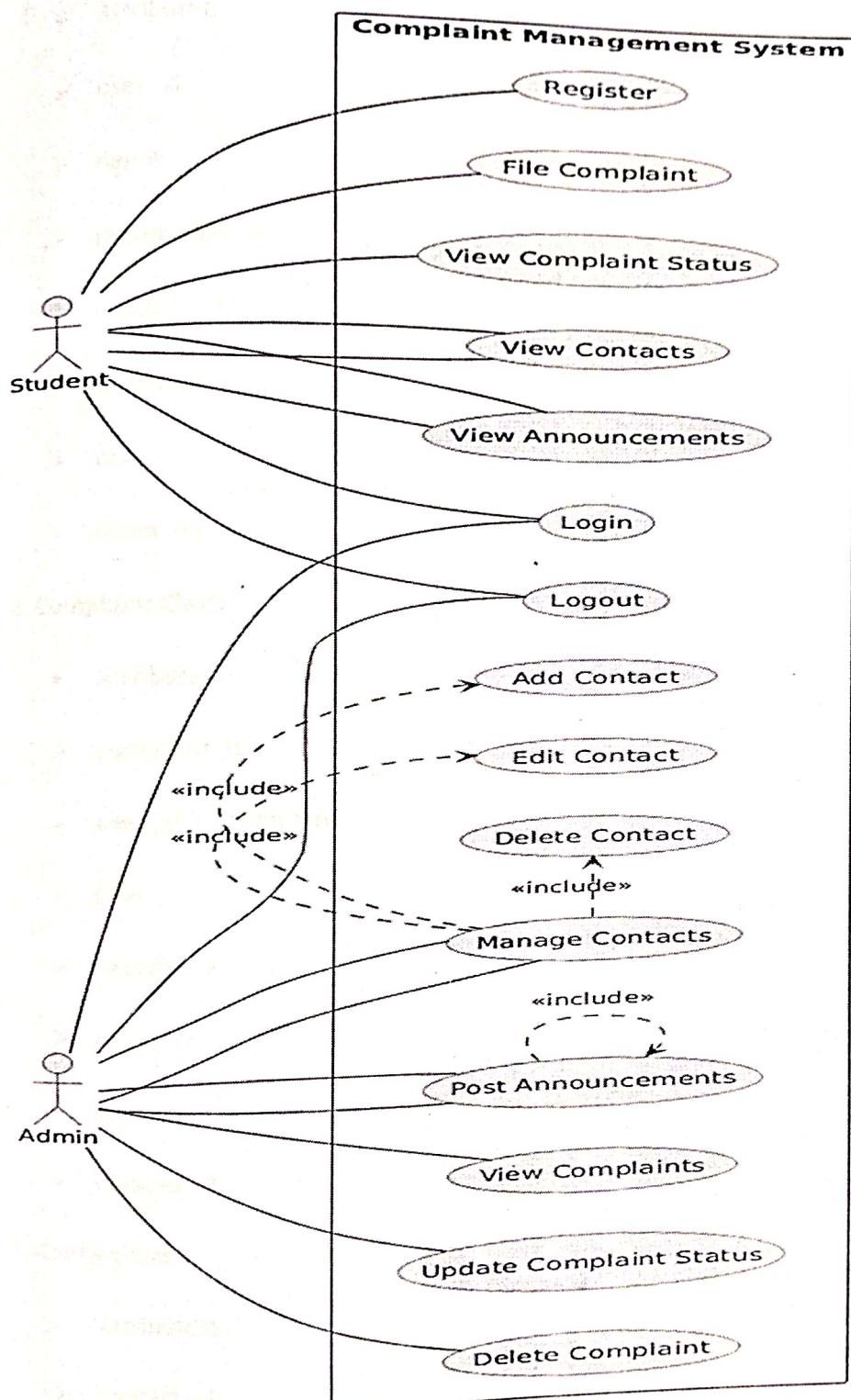
#### Use Cases for Admins:

- Login
- View Complaints
- Update Complaint Status
- Delete Complaint
- Manage Contacts (Add, Edit, Delete)

➤ Post Announcements

➤ Logout

This diagram emphasizes how each user role interacts with specific system features, offering a high-level overview of the system's functionality.



### 3.2.2 Class Diagram

The Class Diagram displays the various entities in the system and their relationships.

#### 1. User Class:

- Attributes:
  - user\_id
  - name
  - registration\_no
  - email
  - password
  - role
  - room\_no

#### 2. Complaint Class:

- Attributes:
  - complaint\_id
  - user\_id (Foreign Key)
  - title
  - description
  - category
  - status
  - created\_at

#### 3. Contact Class:

- Attributes:
  - contact\_id
  - name

- phone\_number

- email

- role

#### 4. Announcement Class:

- Attributes:

- announcement\_id

- admin\_id (Foreign Key)

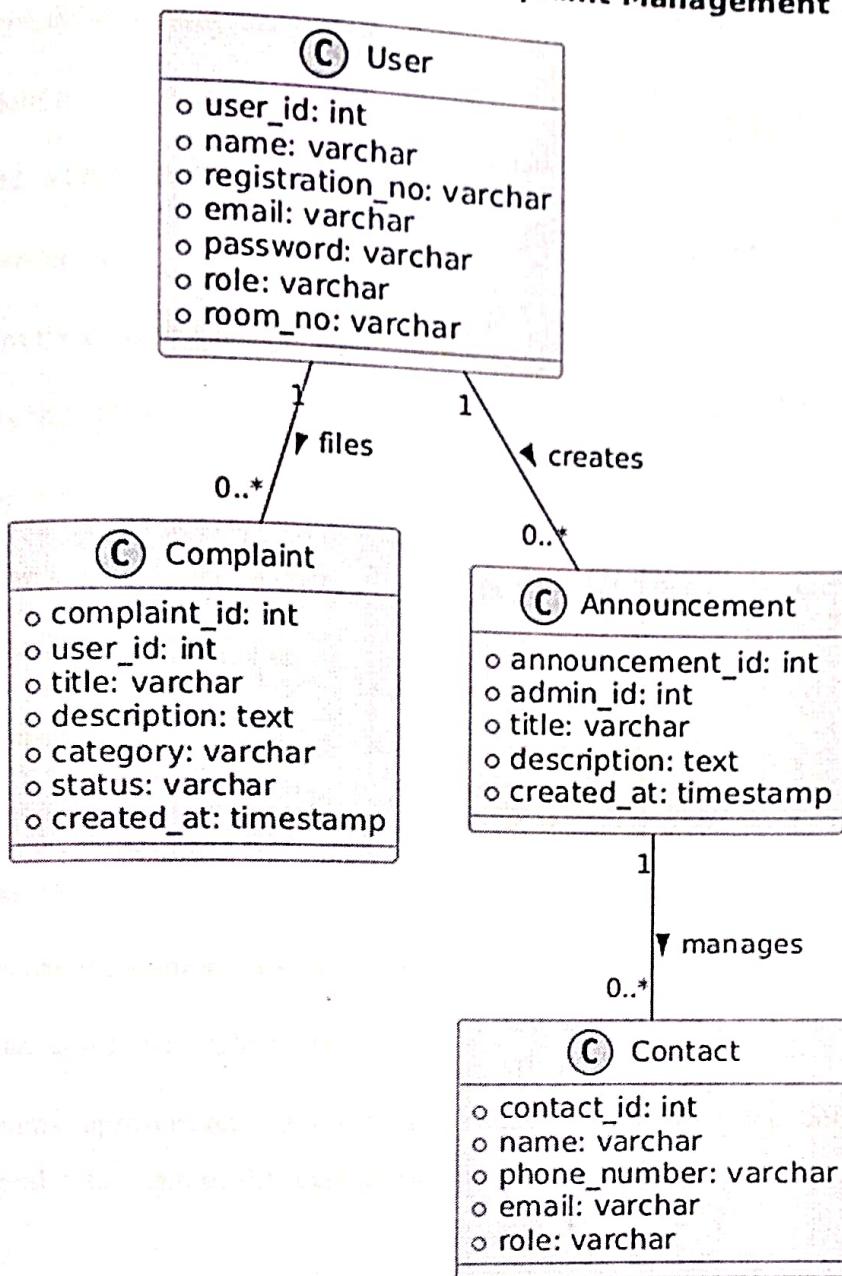
- title

- description

- created\_at

These classes represent the database schema and provide a blueprint for how backend objects are managed in code and stored in the database.

## Class Diagram - FixMyStay Hostel Complaint Management System



### 3.2.3 Activity Diagrams

#### A. Student Files a Complaint:

1. Student logs in.
2. Opens the complaint form.
3. Enters complaint details.
4. System validates inputs.

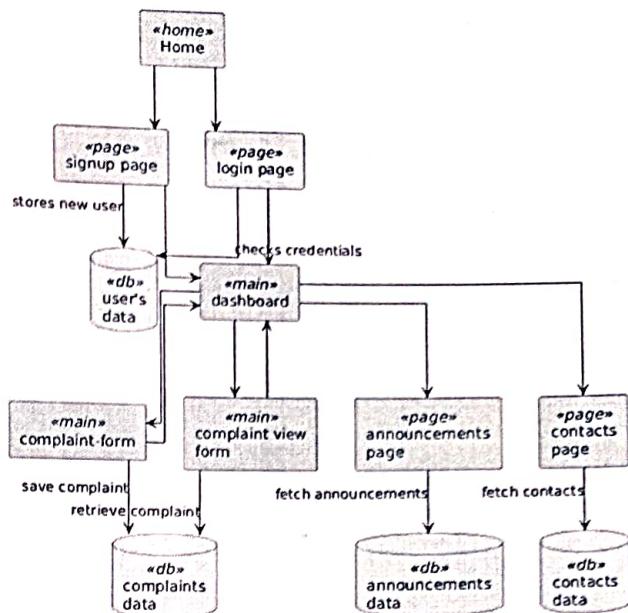
5. Saves complaint in database.
6. Admin updates the status.
7. Student receives notification.

#### B. Admin Adds a Contact:

1. Admin logs in.
2. Opens the contact management module.
3. Clicks "Add Contact".
4. Enters contact information.
5. System saves the new contact.

#### C. Admin Posts an Announcement:

1. Admin logs in.
2. Opens announcements page.
3. Clicks "New Announcement".
4. Enters announcement title and description.
5. System saves and displays it to students.



These diagrams represent real-time workflows and help identify user actions, system responses, and data changes throughout each process.

## **4. IMPLEMENTATION**

## 4. IMPLEMENTATION

The implementation of the FixMyStay - Hostel Complaint Management System is designed to provide a user-friendly, responsive, and efficient experience for both students and hostel administrators. The frontend is developed using HTML, CSS, and JavaScript, ensuring smooth navigation and accessibility across both desktop and mobile devices. The system's interface is structured to be intuitive, making it easy for users to file complaints, track their progress, and access essential hostel-related information.

The system consists of multiple pages, each serving a specific function. The Home Page introduces users to FixMyStay and provides navigation options to log in or sign up. The Login and Signup Pages contain secure forms where users enter their credentials to access the system. Once logged in, users are directed to their respective dashboards. The Student Dashboard allows students to view a summary of their complaints, track complaint statuses, and access the complaint submission form. The Admin Dashboard enables administrators to manage complaints, update their statuses, and ensure their timely resolution.

To facilitate efficient complaint management, the Complaint Form Page allows students to submit complaints by entering details such as title, description, category, and room number. The Complaints List Page displays all submitted complaints in a structured format, categorized by status—Pending, In Progress, and Resolved. Additionally, the system includes a Contacts Page, where students can view hostel contact information while administrators have the ability to add or modify contacts. The Announcements Page serves as a notice board for hostel management to share important updates and notifications.

The frontend design ensures consistency through well-structured CSS styling, making forms, buttons, and tables easy to interact with. JavaScript is utilized to manage form validation, interactive elements, and real-time updates. It ensures users input valid data in forms, enhances dynamic content updates, and provides alerts or messages for successful actions or errors. With this implementation, FixMyStay enhances the efficiency of hostel complaint management by streamlining communication between students and administrators.

## 4.1 SAMPLE CODE

The complaint form page allows students to file a new complaint. The form includes input fields for the title of the complaint, a detailed description of the problem, the complaint category (like Maintenance, Food, Internet, etc.), and their room number. Before submitting, JavaScript validates that none of the fields are left empty. Once the data is validated, it can be submitted, and a confirmation message is shown.

### Complaint-form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>File a Complaint | Hostel Complaint System</title>
    <link rel="stylesheet" href="styles/stylecomp.css">
</head>
<body>
    <nav>
        <h2>FixMyStay</h2>
        <a href="dashboard.html">Dashboard</a>
    </nav>
    <div class="container">
        <h2>File a Complaint</h2>
        <form id="complaintForm">
            <input type="text" id="title" placeholder="Complaint Title" required>
            <textarea id="description" placeholder="Describe the issue" required></textarea>
            <select id="category">
                <option value="Electricity">Electricity</option>
                <option value="Plumbing">Plumbing</option>
                <option value="Internet">Internet</option>
                <option value="Food">Food</option>
                <option value="Other">Other</option>
            </select>
            <input type="text" id="roomNumber" placeholder="Room Number" required>
            <button type="submit">Submit Complaint</button>
        </form>
    </div>
    <script src="scripts/complaint-form.js"></script>
</body>
</html>
```

### Complaint-form.js

```
document.addEventListener("DOMContentLoaded", () => {
  const complaintForm = document.getElementById("complaintForm");
  if (!complaintForm) {
    console.error("Error: Complaint form not found.");
    return;
  }
  complaintForm.addEventListener("submit", async function (event) {
    event.preventDefault();
    let title = document.getElementById("title")?.value.trim();
    let description = document.getElementById("description")?.value.trim();
    let category = document.getElementById("category")?.value.trim();
    let roomNumber = document.getElementById("roomNumber")?.value.trim();
    if (!title || !description) {
      alert("Title and description are required.");
      return;
    }
    category = category || "General";
    roomNumber = roomNumber || "Unknown";
    let userId = localStorage.getItem("userId");
    if (!userId) {
      alert("User not logged in. Please log in first.");
      console.error("Error: User ID not found in localStorage.");
      return;
    }
    let complaintData = {
      userId,
      title,
      description,
      category,
      roomNumber,
      status: "Pending"
    };
    console.log("Sending complaint data:", complaintData);
    try {
      let response = await fetch("http://localhost:3001/api/complaints", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify(complaintData)
      });
      let result = await response.json();
      console.log("Response received:", result);
      if (response.ok) {
        alert("Complaint Submitted Successfully!");
        window.location.href = "complaints.html";
      } else {
    
```

```
        alert(" Error: " + (result.message || "Could not submit complaint."));  
    }  
} catch (error) {  
    console.error(" Error submitting complaint:", error);  
    alert("⚠ Failed to submit complaint. Try again later.");  
}  
});  
});
```

## Dashboard.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Dashboard | FixMyStay</title>  
    <link rel="stylesheet" href="styles/style23.css">  
</head>  
<body>  
    <nav>  
        <div class="nav-left">  
              
            <a href="home.html">FixMyStay</a>  
        </div>  
        <div class="nav-right">  
            <button id="toggleDarkMode">🌙 Dark Mode</button>  
            <button id="logout">Logout</button>  
        </div>  
    </nav>  
    <div class="dashboard-container">  
        <h2>Welcome to FixMyStay</h2>  
        <div id="studentPanel" class="panel">  
            <h3>Dashboard</h3>  
            <a href="complaint-form.html">File a Complaint</a>  
            <a href="complaints.html">View My Complaints</a>  
            <a href="contacts.html">View Contacts ⌂</a>  
        </div>  
        <div id="adminPanel" class="panel" style="display: none;">  
            <h3>Dashboard</h3>  
            <a href="complaints.html">Manage Complaints</a>  
            <a href="contacts.html">Manage Contacts ⌂</a>  
            <a href="announcements.html">Manage Announcements 📢</a>  
        </div>  
        <div class="announcements-section">
```

```

<h3>Latest Announcements</h3>
<ul id="announcementsList">
    <li>Loading announcements...</li>
</ul>
</div>
</div>
<script src="scripts/dashboard.js"></script>
</body>
</html>

```

### dashboard.js

```

document.addEventListener("DOMContentLoaded", async () => {
    const token = localStorage.getItem("token");
    const role = localStorage.getItem("role");
    console.log("Retrieved User Role:", role);
    if (!token) {
        alert("⚠ Unauthorized! Please log in.");
        window.location.href = "login.html";
        return;
    }
    if (role === "admin") {
        document.getElementById("adminPanel").style.display = "block";
        document.getElementById("studentPanel").style.display = "none";
    } else {
        document.getElementById("studentPanel").style.display = "block";
        document.getElementById("adminPanel").style.display = "none";
    }
    document.getElementById("logout").addEventListener("click", () => {
        localStorage.clear();
        window.location.href = "home.html";
    });
    const toggleDarkModeBtn = document.getElementById("toggleDarkMode");
    const body = document.body;
    if (localStorage.getItem("darkMode") === "enabled") {
        body.classList.add("dark-mode");
        toggleDarkModeBtn.innerText = "☀️ Light Mode";
    }
    toggleDarkModeBtn.addEventListener("click", () => {
        body.classList.toggle("dark-mode");
        if (body.classList.contains("dark-mode")) {
            localStorage.setItem("darkMode", "enabled");
            toggleDarkModeBtn.innerText = "☀️ Light Mode";
        } else {
            localStorage.setItem("darkMode", "disabled");
        }
    });
}

```

```

        toggleDarkModeBtn.innerText = "🌙 Dark Mode";
    }
};

await loadAnnouncements();
};

async function loadAnnouncements() {
    const announcementsList = document.getElementById("announcementsList");
    try {
        const response = await fetch("http://localhost:3001/api/announcements");
        const data = await response.json();
        announcementsList.innerHTML = "";
        if (data.length === 0) {
            announcementsList.innerHTML = "<i>No announcements yet.</i>";
            return;
        }
        data.forEach(announcement => {
            const li = document.createElement("li");
            li.textContent = `${announcement.title}: ${announcement.message}`;
            announcementsList.appendChild(li);
        });
    } catch (error) {
        console.error("Error fetching announcements:", error);
        announcementsList.innerHTML = "<i>Failed to load announcements.</i>";
    }
}

```

## 4.2 TEST CASES

Thorough testing is essential to ensure that each module and interaction works as expected. Below are well-structured test cases focusing on the authentication and complaint filing functionalities.

---

### Test Case 1: Login with Valid Credentials

<i>Test Case ID</i>	TC001
<i>Test Case Name</i>	Login with valid credentials
<i>Objective</i>	To verify that a registered user (student or admin) can successfully log in.

**Steps:**

- Open the login page.
- Enter a valid email address.
- Enter the correct password.
- Click the “Login” button.

**Test Data:**

Email: student1@fixmystay.com

Password: password123

<i>Expected Result</i>	<i>The user should be redirected to their respective dashboard.</i>
<i>Actual Result</i>	The user is redirected to the student dashboard.
<i>Status</i>	✓ Pass

**Test Case 2: File Complaint with Valid Data**

Test Case ID	TC002
Test Case Name	<i>Submit a new complaint successfully</i>
Objective	<i>Ensure a complaint is submitted when valid data is entered.</i>

**Steps:**

- Login as student.
- Navigate to “File a Complaint”.
- Enter title, description, category, and room number.
- Click “Submit”.

**Test Data:**

Title: "Broken fan"

Description: "The fan in my room is not working."

Category: "Electricity"

Room Number: "B201"

<i>Expected Result</i>	<i>Complaint is submitted and status is set to “Pending”.</i>
<i>Actual Result</i>	Complaint submitted successfully, status is “Pending”.

Status

✓ Pass

### Test Case 3: Leave Required Field Empty

Test Case ID TC003

Test Case Name	Error on submitting without title or description
Objective	Validate that the form throws an alert when fields are missing.

Steps:

- Open complaint form.
- Leave title or description empty.
- Click "Submit".

Expected Result Alert should appear stating required fields are missing.

Actual Result	Alert appears: "⚠ Title and description are required."
Status	✓ Pass

### Test Case 4: Submit Complaint Without Login

Test Case ID TC004

Test Case Name	Prevent complaint submission if not logged in
Objective	Ensure users cannot submit complaints without logging in.

Steps:

- Open complaint form directly without logging in.
- Enter valid complaint data.
- Click "Submit".

Expected Result Alert: "User not logged in. Please log in first."

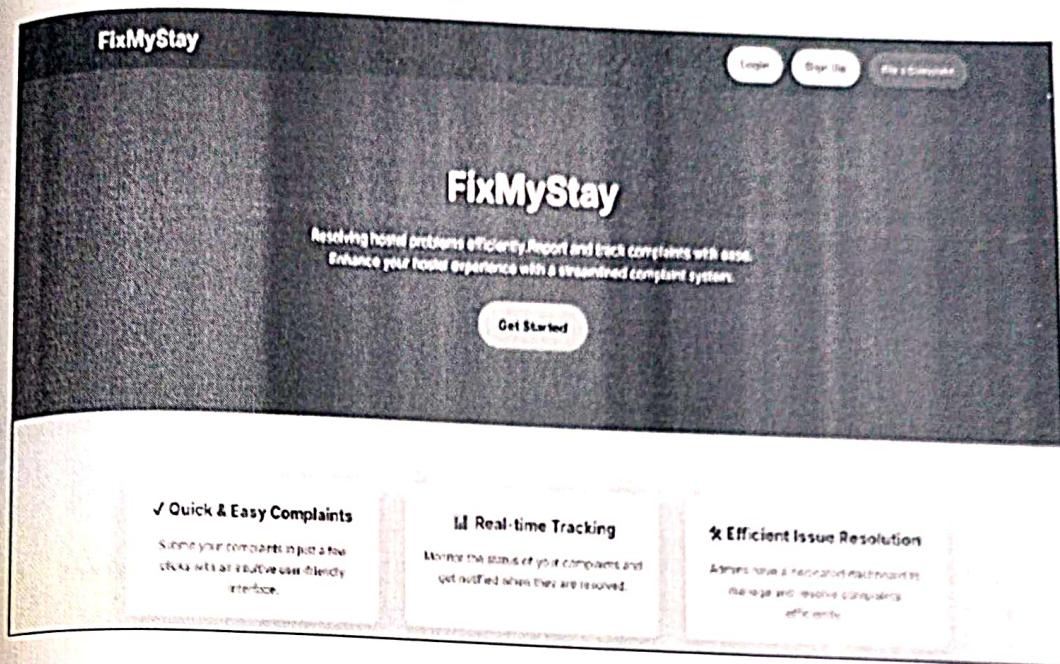
Actual Result	Alert shown. Complaint not submitted.
Status	✓ Pass

## **5. RESULTS**

## 5.RESULTS

### 5.1 OUTPUT SCREENS

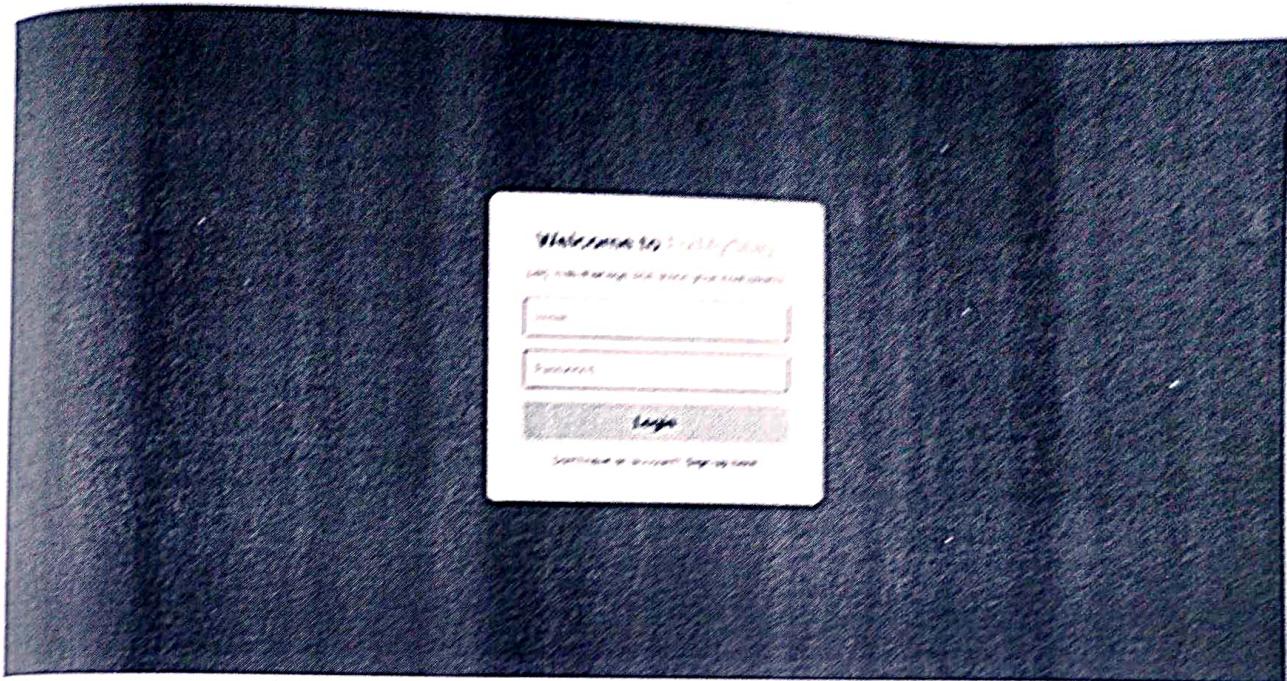
- The Home page



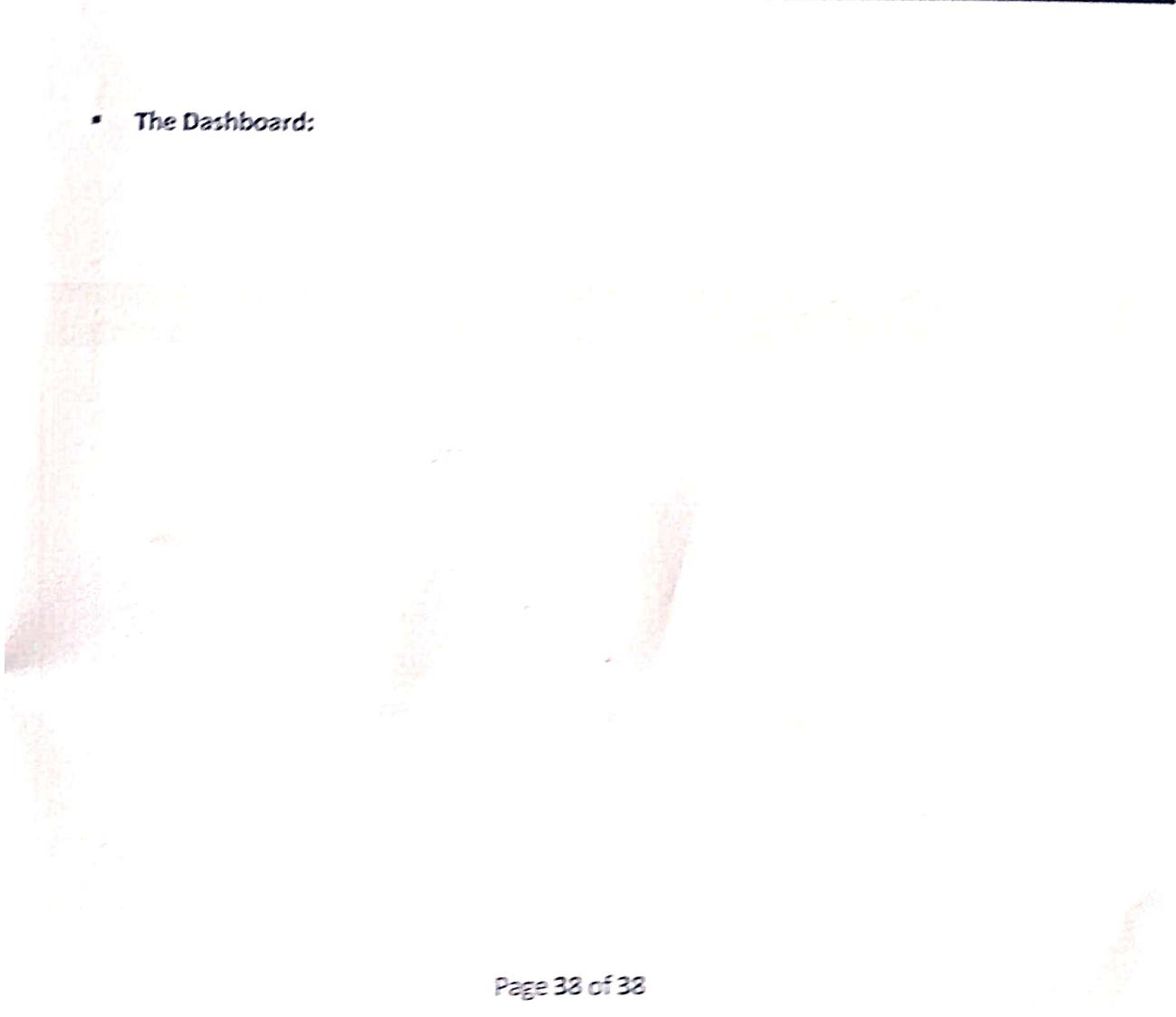
- The signup page:

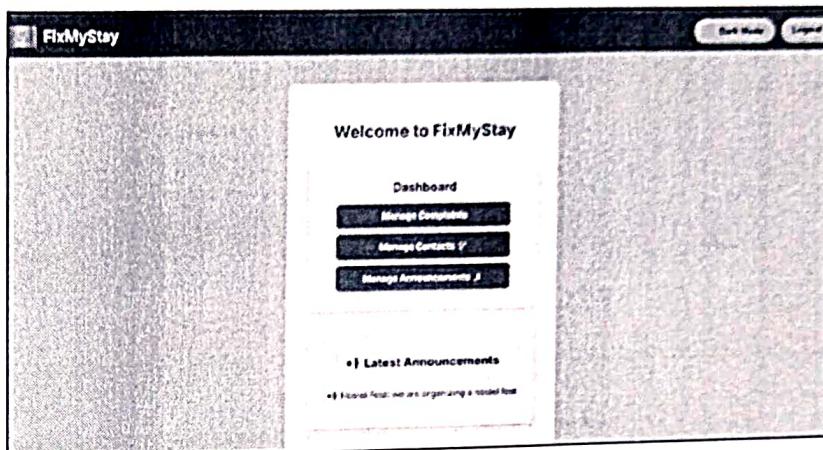
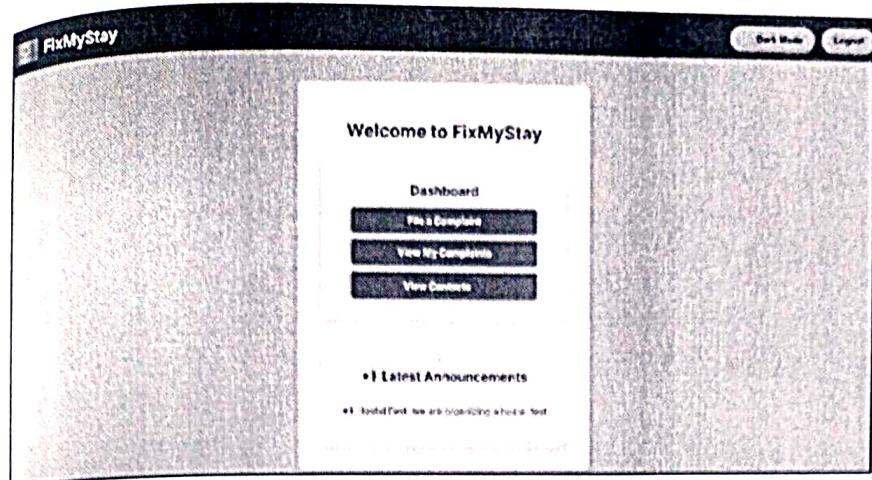
The screenshot shows the "Create an Account" sign-up page. It features a logo with the word "fixmystay" and a "Sign Up" button. The form fields include "Full Name", "Email Address", "Password", and "User". Below the form is a "Sign Up" button and a link for "Already have an account? Log in".

- The login page:



- The Dashboard:





- File a complaint page:

FixMyStay

File a Complaint

Complaint Title

Describe the issue

Category: Electricity

Room Number

- View complaint page:
- The announcements page:

**Hostel Announcements**

[Edit Info](#) [Edit Message](#) [Add](#)

**Hostel Fest**  
we are organizing a hostel fest  
Posted on: 4/3/2025, 7:35:44 PM

[Details](#) [Back to Dashboard](#)

**FixMyStay** [Dashboard](#)

**Complaint List**

Filter by Category: All

ID	Title	Description	Category	Room	Status	Action
13	Chicken	Chicken quality is decreased from past two weeks.	Food	5	Resolved	

**FixMyStay** [Dashboard](#)

**Complaint List**

Filter by Category: All

ID	Title	Description	Category	Room	Status	Action
13	Lights	The lights in the washroom are not working.	Electrical	3	Resolved	
11	No Quality	The quality of the food has decreased.	Food	4	Resolved	
10	Chicken	Chicken quality is decreased from past two weeks.	Food	1	Resolved	
10	Slow net	We are not getting a proper speed of internet these days. As we are having exam it is better to have a faster internet connection.	Internet	9	Pending	

■ The contacts page:

The screenshot shows a web application interface for managing contacts. At the top left is the logo "FixMyStay". At the top right is a link "→ Back to Dashboard". The main title is "Contact Information". Below it, there is a section for "Warden" with the following details:

- Phone: 656162438
- Email: warden@fixmystay.com

Below these details are two buttons: "Edit" and "Delete". At the bottom of the contact card is a large "Add Contact" button.

## **6.CONCLUSION**

## 6.CONCLUSION

The FixMyStay – Hostel Complaint Management System is an innovative web-based platform created to address the inefficiencies of traditional, manual complaint handling systems in hostels. It replaces outdated methods like physical registers or verbal complaints with a digital, streamlined solution that ensures accountability, transparency, and real-time tracking of issues faced by hostel residents.

By enabling students to file complaints anytime, from anywhere, the system provides accessibility and convenience. Each complaint filed is tagged with relevant details such as title, description, category (like Internet, Food, Electricity, etc.), and room number, which helps administrators quickly understand and prioritize them. Complaints are automatically categorized by status — Pending, In Progress, or Resolved — allowing students to track progress without needing repeated follow-ups. This fosters a transparent communication bridge between students and management.

On the administrative side, FixMyStay offers a centralized dashboard where hostel authorities can efficiently monitor all registered complaints. They can update statuses, assign specific tasks to technicians or housekeeping staff, and leave remarks for future reference. These features empower administrators to streamline their workflows, respond promptly, and maintain an organized log of all activities. This enhances service delivery and overall hostel management quality.

One of the key strengths of FixMyStay lies in its modular and scalable architecture. Its components — including complaint handling, authentication, announcement management, and contact management — are all designed for extensibility. This structure makes it easy to introduce new features and scale the system as the number of users or hostels increases.

Additionally, FixMyStay introduces student engagement through a clean, intuitive UI/UX, and built-in functionalities like dark mode, announcements, and role-based dashboards. It elevates the user experience for both students and administrators alike.

Scope for Future Enhancements:

- **Mobile Application Support:** To increase accessibility, especially for on-the-go users, a dedicated mobile app could be developed for Android and iOS platforms.

- **Automated Notifications:** Integration of real-time email or SMS alerts to notify students when their complaint status is updated.
- **AI-Powered Categorization:** Use of machine learning algorithms to auto-suggest complaint categories or detect common issues.
- **Feedback & Rating System:** After resolution, students can rate the response and give feedback to help improve service quality.
- **Multi-Hostel Support:** Extend the platform to support multiple hostels or campuses under a centralized admin portal.

In conclusion, FixMyStay bridges the communication gap between students and hostel authorities by offering a reliable, user-friendly, and organized approach to managing complaints. It not only saves time but also enhances trust, accountability, and the overall living experience in hostels. With future upgrades, the system has the potential to evolve into a comprehensive hostel management suite.

## 7. REFERENCES

Project Link: <https://github.com/saathvikb2005/hostel-complaint-system>