

HOSTELIZE – A HOSTEL COMPLAINTS MANAGEMENT PLATFORM

A MINI-PROJECT REPORT

Submitted by

TINU A V

211701057

in partial fulfillment for the course

CD19643- Web Essentials

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND DESIGN

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI - 602 105

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE CHENNAI

602105

BONAFIDE CERTIFICATE

Certified that this project report **“HOSTELIZE – A HOSTEL COMPLAINT MANAGEMENT PLATFORM”** is the bonafide work of **“TINU A V (211701057)”** who carried out the project work for the subject **CD19643- Web Essentials** under my supervision.

SIGNATURE

Prof. Uma Maheshwar Rao
Head of the Department

Associate Professor

Department of Computer Science and
Design

Rajalakshmi Engineering College
Chennai - 602105

SIGNATURE

Dr.N.Duraimurugan,M.Tech.,Ph.D.,
Supervisor

Assistant Professor

Department of Computer Science and
Engineering

Rajalakshmi Engineering College
Chennai - 602105

Submitted to Project and Viva Voce Examination for the subject CD19643-
Web Essentials held on _____.

Internal Examiner

External Examiner

ABSTRACT

This project entails the development of a user-friendly website named “Hostelize” catering to hostel students' needs by facilitating complaint registration with the added feature of attaching images. The platform is designed to streamline the communication between students and administrative personnel responsible for managing hostel affairs. Students can easily register complaints regarding various hostel-related issues, such as maintenance, facilities, or security, and provide visual evidence by attaching images directly within the complaint submission form. This feature enhances the clarity and specificity of the reported issues, enabling administrators to better understand and address them promptly. By integrating image attachment functionality, the platform empowers students to express their concerns vividly while enabling administrators to gain deeper insights into the nature and severity of reported problems. Additionally, the platform offers a secure environment for students to submit complaints, maintaining confidentiality and privacy throughout the process. This innovative approach not only fosters a more transparent and responsive complaint resolution process but also contributes to creating a safer, more comfortable living environment within the hostel premises.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr.S.Meganathan, B.E, F.I.E.**, our Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.**, and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.**, for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to our **Prof. Uma Maheshwar Rao** Associate Professor and Head of the Department of Computer Science and Design for his guidance and encouragement throughout the project work. We convey our sincere thanks to our internal guide and Project Coordinator, **Dr.N.Duraimurugan, M.Tech., (PhD).**, Department of Computer Science and Engineering, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

TINU A V (211701057)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	LIST OF FIGURES	vi
1.	INTRODUCTION	1
	1.1 OBJECTIVE	
2.	LITERATURE REVIEW	3
3.	PRESENT TECHNOLOGY	4
4.	3.1 LIMITATIONS	
4.	PROPOSED TECHNOLOGY	6
	4.1 SEQUENCE DIAGRAM	
	4.2 USECASE DIAGRAM	
	4.3 SCHEMATIC DIAGRAM	
	4.4 ADVANTAGES	
5.	RESULT AND CONCLUSION	13
	5.1 OUTPUT SCREENSHOTS	
6.	CONCLUSION	19
	REFERENCES	

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE.NO
4.1	SEQUENCE DIAGRAM	9
4.2	USECASE DIAGRAM	10
4.3	SCHEMATIC DIAGRAM	11
5.1	OUTPUT SCREENSHOTS	15

CHAPTER 1

INTRODUCTION

Hostelize is a comprehensive web-based solution tailored to meet the diverse needs of hostel residents, offering a streamlined platform for registering and managing complaints effectively. With a growing emphasis on digital connectivity and convenience, Hostelize bridges the gap between students and administrative authorities, revolutionizing the way hostel-related issues are addressed. This innovative website empowers students to voice their concerns with ease, utilizing features such as image attachment to provide detailed evidence alongside their complaints. By harnessing the power of technology, Hostelize enhances transparency, efficiency, and accountability within hostel management, fostering a more conducive living environment for residents.

Through Hostelize, students gain access to a user-friendly interface where they can submit complaints regarding various aspects of hostel life, from maintenance and facilities to security and hygiene. The incorporation of image attachment functionality allows students to provide visual documentation, ensuring a clearer understanding of the reported issues and enabling administrators to prioritize and address them promptly. Administrators, equipped with robust tools for complaint management, can efficiently track, analyze, and resolve complaints, thereby enhancing overall hostel administration.

Hostelize not only facilitates smoother communication between students and administrators but also promotes a culture of responsiveness and accountability within hostel management. By offering real-time updates on complaint status, Hostelize ensures that students remain informed throughout the resolution process, instilling confidence in the effectiveness of the administrative framework. As a result, Hostelize emerges as a vital tool in enhancing the hostel living experience, fostering a sense of community and well-being among residents.

1.1 OBJECTIVE

The goal is to create a web-based platform for efficient complaint management in hostels, enhancing communication between students and administrators. This system will streamline the complaint resolution process, ensuring timely action on reported issues, improving student satisfaction and overall hostel experience. The platform will be transparent and accountable, providing administrators with a user-friendly dashboard for managing complaints, assigning tasks, and monitoring progress. The ultimate goal is to promote student well-being and community cohesion within hostel premises. The objective of the project is to develop a comprehensive website for hostel management that enables students to easily register complaints regarding hostel facilities or services and allows administrators to efficiently manage and address these complaints. The website aims to streamline the complaint submission process, enhance communication between students and administrators, provide transparency in the complaint resolution process, and ultimately improve the overall living experience for hostel residents. Additionally, the project seeks to implement robust features for storing student details, tracking complaint status, and generating reports to facilitate effective hostel management.

CHAPTER 2

LITERATURE REVIEW

In the academic context, academic advancement can be of different concerns to strengthen the social and productive education system., for an effective educational system to take place, t certain issues in the academic environment must be properly addressed. Therefore, with the advancement of technology, internet users are increasing exponentially, it would benefit from technology in many things, including the complaints management system in universities. This project identifies a set of options that can be used to manage and collect student complaints. It is possible to design and implement an online complaints management system to benefits the student as well as the university so that the student can share the problems that student faces at the university and the university will be able to solve the student's problems and provide an appropriate educational environment for all students. This final year project will focus on the development of a complaints management system on the Internet will be for universities that do not have this system, especially universities in Yemen due to the lack of this system. This system works on Monitor student complaints, improve the database, and enhance the effectiveness, safety, and security of the system. The complaints handling system is easily accessible and presented to complainants (the students). The editing software uses brackets, also PHP as the programming language, as well as MySQL as the database. However, the importance of this project will increase more benefits for the students and the University.

CHAPTER 3

PRESENT TECHNOLOGY

HTML (Hypertext Markup Language): For structuring the content of web pages.

CSS (Cascading Style Sheets): For styling the appearance of web pages.

JavaScript: For implementing interactive features and client-side logic.

Front-end Frameworks/Libraries: Such as React.js, Angular, or Vue.js may be used to streamline development and enhance user experience.

Server-side Scripting Languages: Such as PHP, Python (with frameworks like Django or Flask), Ruby (with Ruby on Rails), or Node.js for handling server-side logic and processing user requests. Database Management Systems (DBMS): Like MySQL, PostgreSQL, MongoDB, or Firebase for storing and managing complaint data.

Server Environment: Such as Apache or Nginx for serving web pages and handling HTTP requests. Back-end Frameworks: Such as Express.js (for Node.js), Flask (for Python), or Laravel (for PHP) for building robust back-end applications more efficiently.

APIs (Application Programming Interfaces): Used for communication between the front-end and back-end systems, allowing data exchange and interaction.

Authentication and Authorization: Techniques like JWT (JSON Web Tokens) or OAuth may be implemented to secure user authentication and authorization processes.

Version Control Systems: Such as Git for managing codebase changes and collaboration among developers. Testing Frameworks: Like Jest (for JavaScript), PHPUnit (for PHP), or PyTest (for Python) for ensuring code reliability and quality.

DevOps Tools: Such as Docker for containerization, Jenkins or Travis CI for continuous integration, and deployment tools like Heroku or AWS for hosting the application. Implementation of HTTPS to ensure secure data transmission.

Input validation and sanitization to prevent SQL injection, XSS (Cross-Site Scripting), and other security vulnerabilities. User authentication mechanisms to protect sensitive user data and ensure authorized access to complaint records.

3.1 LIMITATIONS

Learning Curve: Some of the technologies mentioned, such as React.js, Angular, Django, or Node.js, have steep learning curves, especially for developers who are new to web development. Mastering these technologies may require significant time and effort.

Performance Overhead: front-end frameworks like React.js or Angular can introduce performance overhead, especially for smaller projects where the complexity of these frameworks may not be necessary. Overuse of JavaScript libraries and frameworks can also lead to slower page load times and decreased performance.

Complexity: Integrating multiple technologies and frameworks increases the complexity of the project, making it more challenging to maintain and debug. Additionally, compatibility issues may arise when combining different libraries and frameworks.

Scalability: While many of the technologies mentioned are scalable, achieving scalability requires careful architecture design and optimization. Inefficient database queries, server configurations, or application architecture can hinder scalability as the system grows.

Security Concerns: Implementing user authentication and authorization mechanisms is crucial for protecting sensitive user data. However, improper implementation or misconfiguration of authentication mechanisms can lead to security vulnerabilities such as account hijacking, session fixation, or unauthorized access to complaint records.

Maintenance Overhead: Regular maintenance and updates are necessary to keep the system secure and up-to-date with the latest security patches and framework updates. This can be time-consuming and may require dedicated resources.

Testing Challenges: Testing a complex web application with multiple layers (front-end, back-end, database) and dependencies can be challenging. Ensuring adequate test coverage and writing comprehensive test cases require careful planning and execution.

CHAPTER 4

PROPOSED TECHNOLOGY

HTML (Hypertext Markup Language):

HTML is the standard markup language used to create the structure and layout of web pages. It consists of a series of elements, each representing different types of content such as headings, paragraphs, images, links, forms, and more. HTML provides the foundation for building web pages and is essential for creating a structured document that browsers can interpret and display to users.

CSS (Cascading Style Sheets):

CSS is a style sheet language used to define the visual presentation and layout of HTML elements on a web page. With CSS, web developers can control various aspects of the appearance of elements, such as colors, fonts, spacing, borders, and positioning. CSS allows for the separation of content from presentation, making it easier to maintain and update the visual style of a website across multiple pages.

JavaScript:

JavaScript is a versatile programming language commonly used for adding interactivity and dynamic behavior to web pages. It runs on the client-side within the web browser and can be used to manipulate HTML elements, handle user interactions, perform form validation, create animations, fetch data from servers asynchronously (using AJAX), and much more. JavaScript is essential for creating engaging and interactive web experiences.

PHP (Hypertext Preprocessor):

PHP is a server-side scripting language primarily used for web development. It is embedded within HTML code and executed on the server to generate dynamic web content. PHP is commonly used for tasks such as processing form data, interacting with databases, generating dynamic web pages, and managing user sessions. It is widely supported by web hosting providers and is commonly used in conjunction with databases like MySQL to build dynamic and data-driven websites.

INTEGRATING IMAGE UPLOADS :

When integrating the functionality to add images while registering complaints, the system allows users to upload images directly from their devices as part of the complaint submission process. Upon selecting the desired image files, the system facilitates their seamless transfer to the server. Once received, the images undergo processing, including validation of file types and sizes, to ensure compliance with system requirements.

Subsequently, the validated images are stored securely within the database, using appropriate techniques such as encoding the images into a binary format (e.g., BLOB or BYTEA) to preserve their integrity and maintain efficient storage. Each uploaded image is associated with its respective complaint, with metadata linking it to the corresponding complaint ID for seamless retrieval and referencing.

This approach streamlines the management of images associated with complaints, eliminating the need for separate storage solutions and ensuring data consistency within the centralized database. Moreover, storing images in the database simplifies data access and retrieval, facilitating seamless integration with other system functionalities such as viewing complaint details and generating reports.

Overall, by enabling users to include images alongside their complaints and storing them directly in the database, the system enhances the comprehensiveness and visual context of the complaint submission process, ultimately contributing to a more efficient and effective complaint management system.

SOFTWARES UTILIZED:

XAMPP:

XAMPP stands as a versatile solution for local web server needs, crafted by Apache Friends. It amalgamates essential components like Apache HTTP Server, MariaDB (formerly MySQL), PHP, and Perl. This comprehensive package empowers developers to swiftly establish a local web server environment on their machines. This local setup offers a sandboxed space for testing web applications prior to their deployment on live servers, thus minimizing the risk of errors and issues reaching the production stage. XAMPP simplifies the intricate process of installing and configuring these server-side components, making it an invaluable asset for developers

seeking a seamless development experience. By leveraging XAMPP, developers can effortlessly experiment with various configurations, troubleshoot issues, and fine-tune their applications in a controlled environment, ultimately enhancing the quality and reliability of their web projects.

Visual Studio Code:

Visual Studio Code, commonly referred to as VS Code, emerges as a lightweight yet potent code editor fostered by Microsoft. Offering a rich array of features and functionalities, it caters to a diverse array of programming languages and development workflows. From its intuitive interface to its extensive ecosystem of extensions, VS Code is engineered to boost developer productivity and efficiency. Syntax highlighting, code completion, and integrated debugging capabilities streamline the coding process, while version control integration facilitates seamless collaboration among team members. VS Code's versatility and adaptability make it a favored choice among developers for a myriad of tasks, ranging from crafting frontend code in HTML, CSS, and JavaScript to developing server-side logic in PHP and Python. Its cross-platform compatibility further extends its reach, enabling developers to wield its power across different operating systems and development environments.

Google Chrome:

Google Chrome emerges as a leading force in the realm of web browsers, renowned for its speed, security, and versatility. Equipped with robust support for modern web technologies, including HTML, CSS, JavaScript, and WebGL, Chrome provides an optimal platform for previewing and testing web applications during development. Its Developer Tools furnish developers with an arsenal of features for debugging, testing, and optimizing web applications, empowering them to diagnose issues, analyze performance metrics, and fine-tune their creations with precision. Chrome's extensive ecosystem of extensions further augments its capabilities, enabling developers to tailor their browsing experience to suit their specific needs and preferences. As a cornerstone of web development workflows, Chrome plays an indispensable role in the iterative process of building and refining web applications, helping developers bring their visions to life with unparalleled efficiency and effectiveness.

4.1 SEQUENCE DIAGRAM

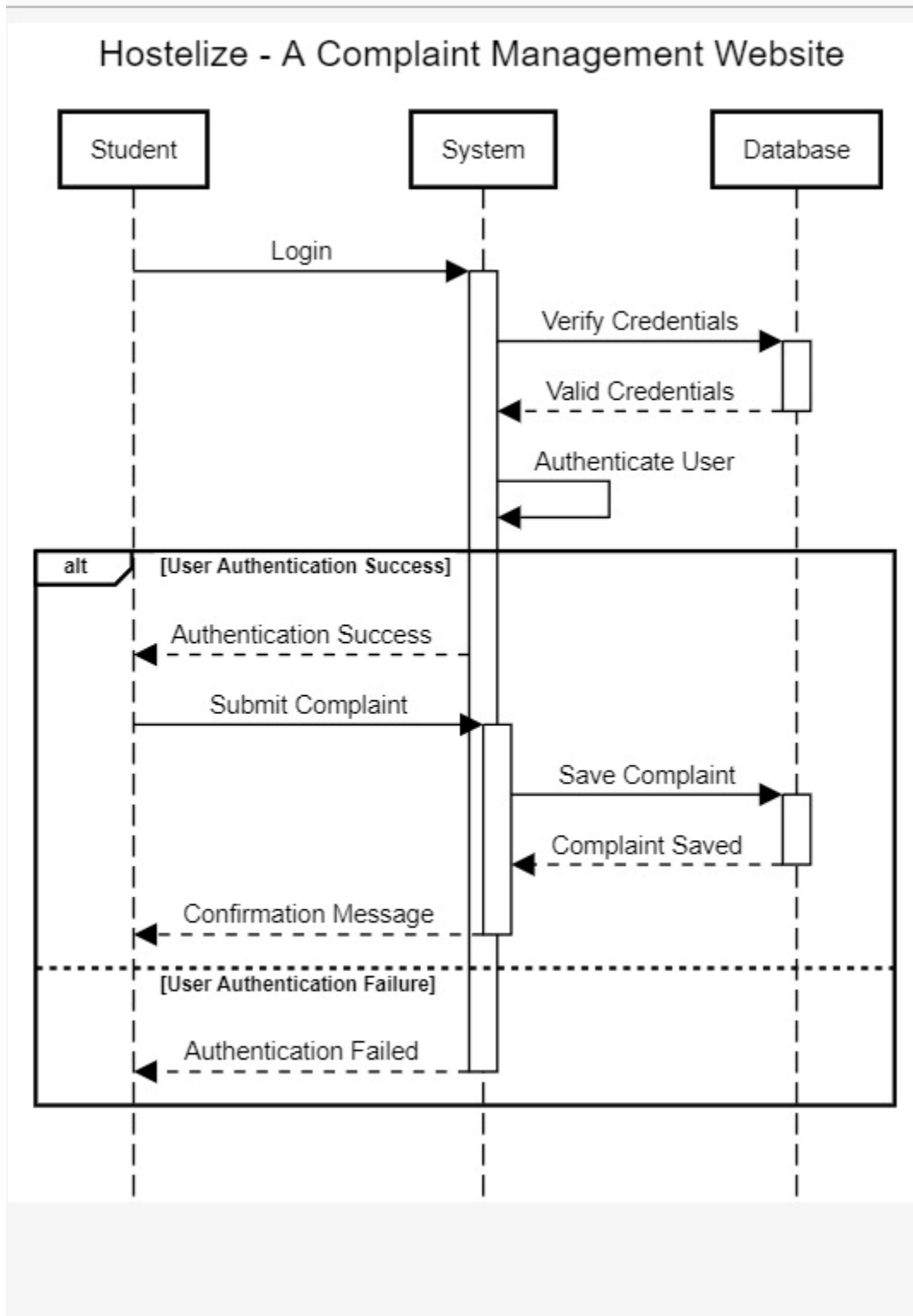


Fig 4.1 Sequence diagram

4.2 USECASE DIAGRAM

A use case diagram is a type of behavioral diagram in Unified Modeling Language (UML) that illustrates the interactions between actors (users or external systems) and a system to achieve specific goals or functionalities. It provides a high-level view of the system's behavior and the ways in which users interact with it.

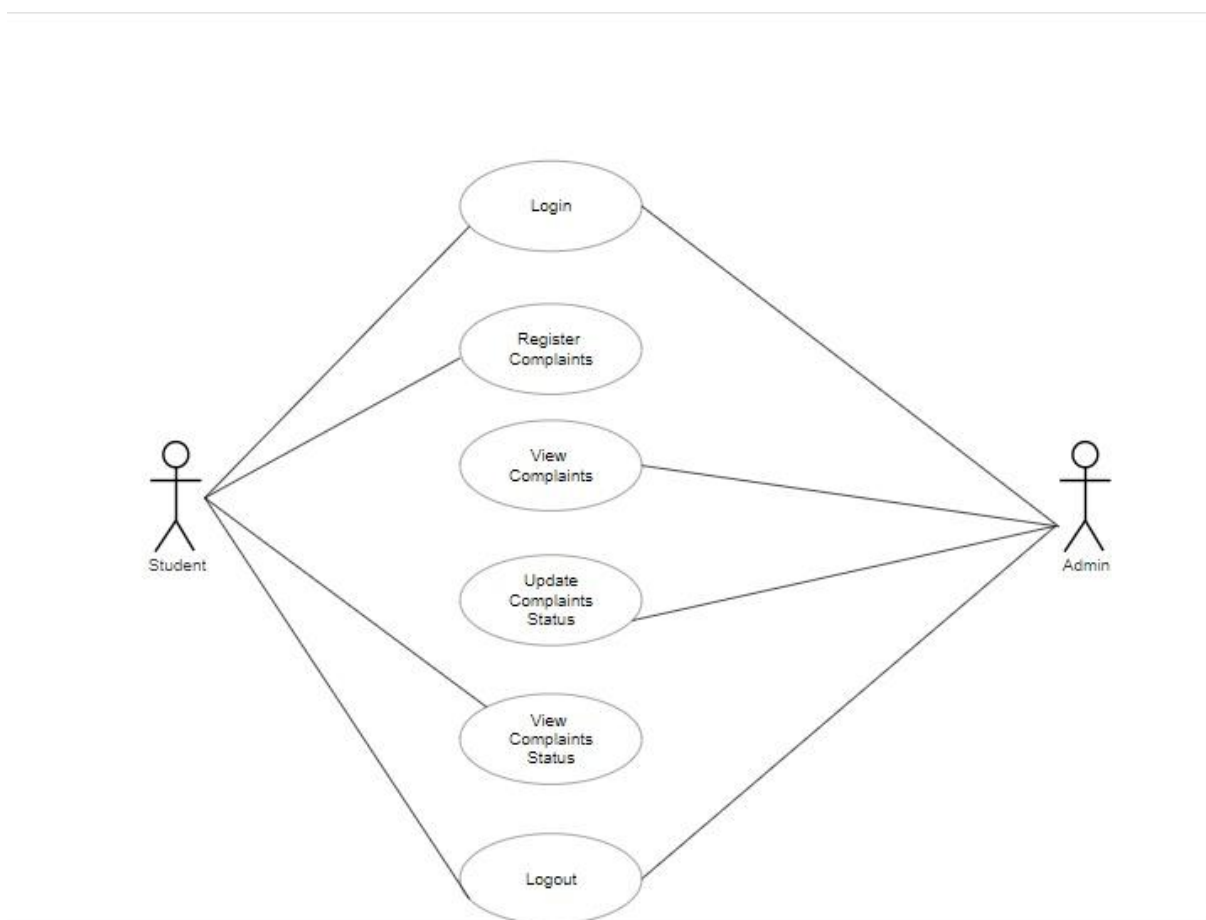


Fig 4.2 Usecase Diagram

4.3 SCHEMATIC DIAGRAM:

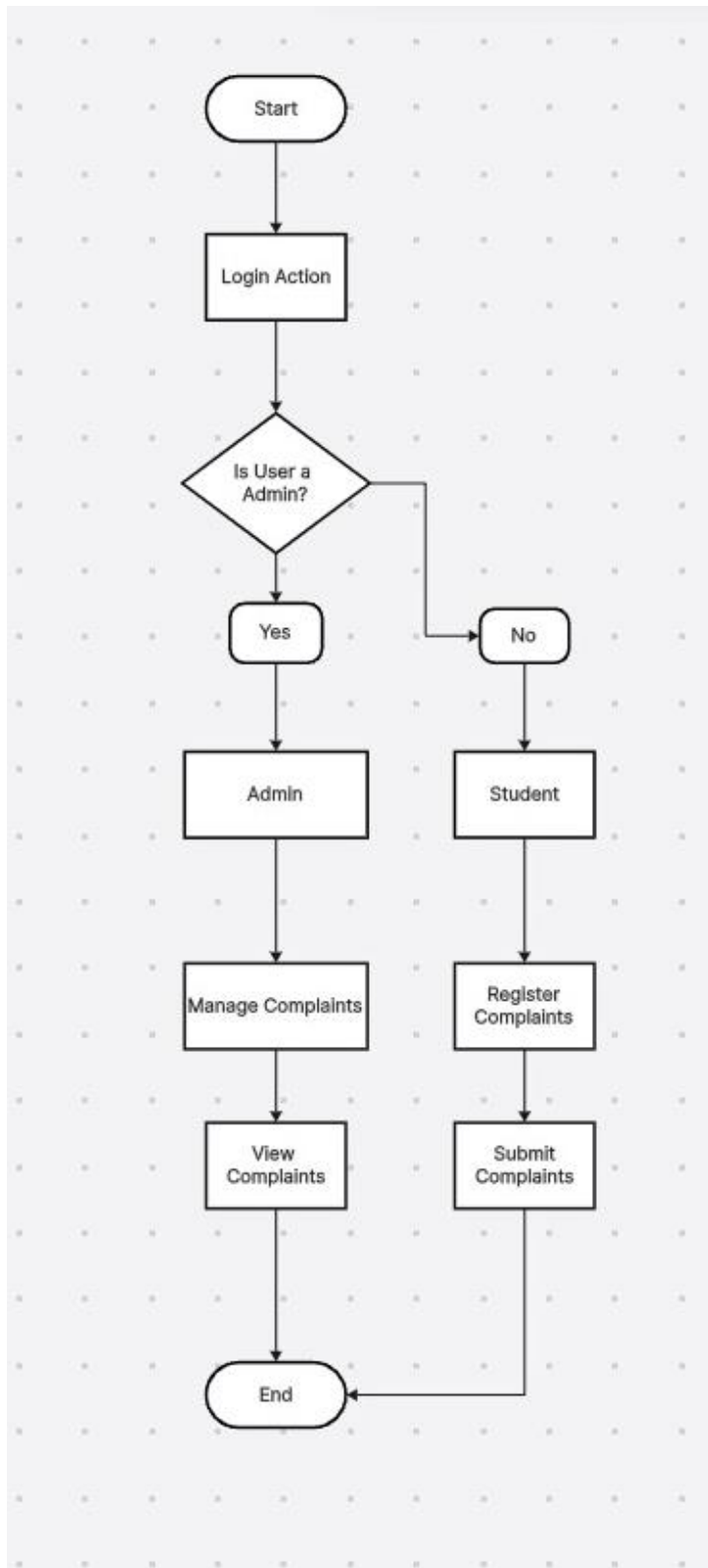


Fig.4.3 Schematic Diagram

4.4 ADVANTAGES

Improved Communication: The website provides a centralized platform for hostel students to report complaints, fostering better communication between students and administrators. This can lead to quicker resolution of issues and a more positive living environment in the hostel.

Efficient Complaint Management: The system streamlines the process of registering, tracking, and resolving complaints, allowing administrators to manage complaints more efficiently. This can result in faster response times and increased satisfaction among students.

Transparency and Accountability: By providing students with visibility into the status of their complaints, the website promotes transparency and accountability in the complaint resolution process. Students can track the progress of their complaints and feel reassured that their concerns are being addressed.

Data Management: The system facilitates the collection and management of data related to student complaints, enabling administrators to identify recurring issues, track trends, and make informed decisions to improve hostel management.

Accessibility: The website offers 24/7 accessibility, allowing students to register complaints at any time, even outside of regular office hours. This ensures that no complaints are overlooked and that issues can be addressed promptly.

Scalability: The project can be scaled to accommodate the needs of larger hostels or expanded to include additional features, such as feedback mechanisms, event notifications, or community forums, enhancing its utility and value over time.

User Satisfaction: By providing a user-friendly interface and efficient complaint resolution process, the website can enhance overall user satisfaction among hostel students and administrators, contributing to a positive experience for all stakeholders.

CHAPTER 5

RESULT AND DISCUSSION

Presentation of Results:

Provide an overview of the implemented website, including its features, functionalities, and user interface design.

Present quantitative data on the number of complaints registered, average resolution time, and user engagement metrics (e.g., number of active users, frequency of complaint submissions).

Include qualitative feedback from users, such as students' satisfaction with the complaint submission process and administrators' experiences with managing complaints.

Analysis of Results:

Discuss the impact of the website on communication between students and administrators. Evaluate whether the website has facilitated better communication and transparency in the complaint resolution process.

Analyze the efficiency of complaint management with the implemented system. Compare the average resolution time before and after the implementation of the website and identify any bottlenecks or areas for improvement.

Evaluate the accessibility of the website and its usability for both students and administrators. Consider factors such as ease of complaint submission, navigation, and responsiveness across different devices.

Discuss the scalability of the system and its ability to accommodate increasing numbers of users and complaints. Identify any challenges encountered in scaling the system and propose potential solutions.

Comparison with Existing Systems or Practices:

Compare the performance of the implemented website with existing systems or manual practices for managing hostel complaints. Highlight any improvements in efficiency, transparency, or user satisfaction achieved with the website.

Discuss how the website addresses limitations or shortcomings of previous complaint management systems and contributes to enhancing hostel management processes.

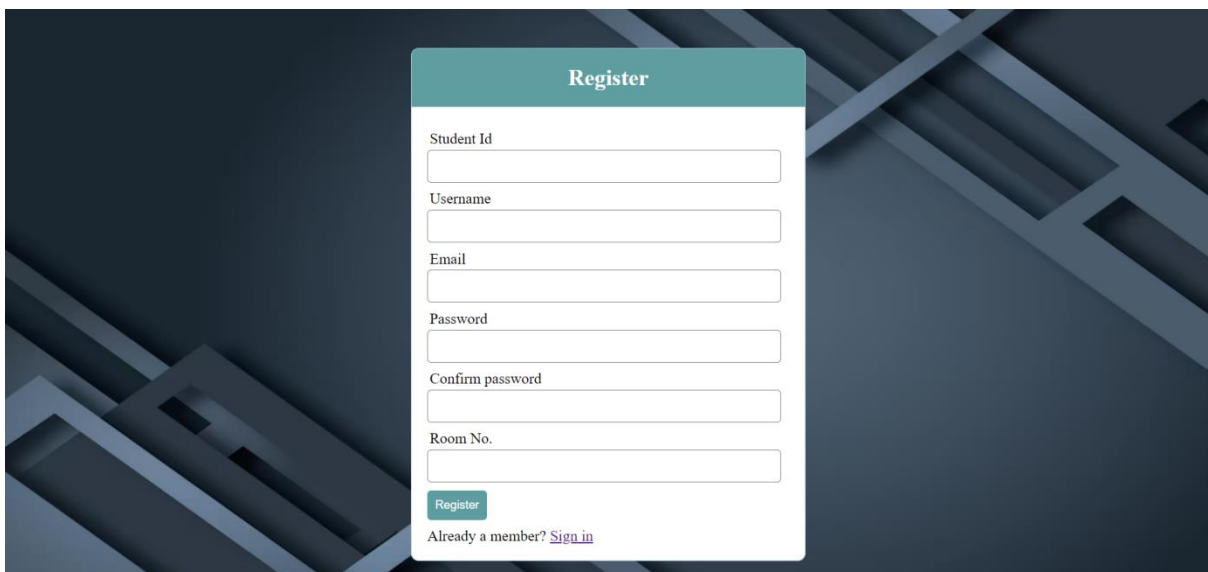
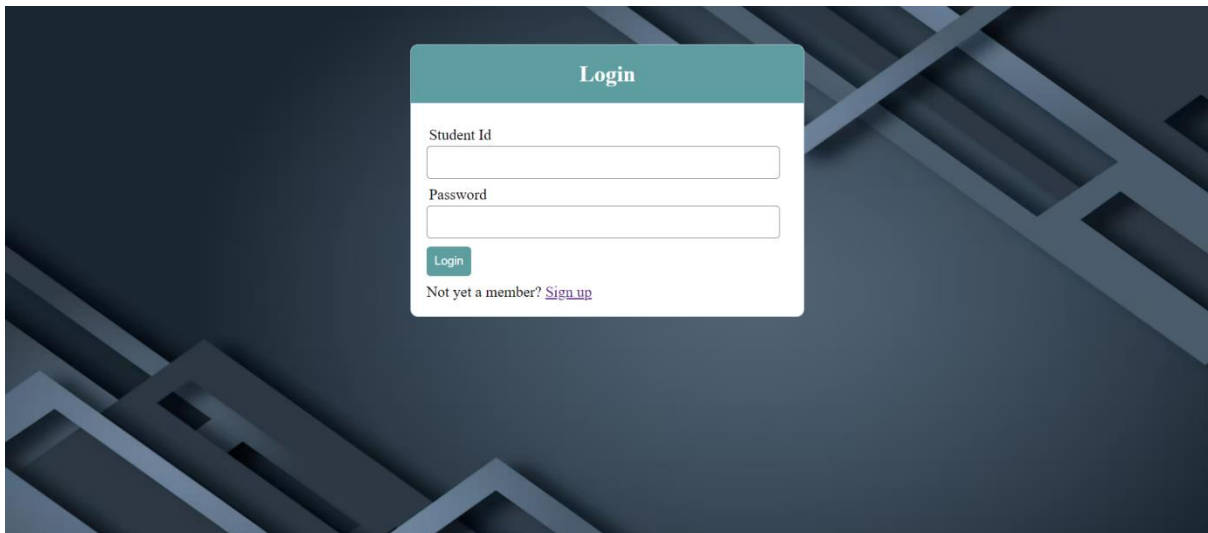
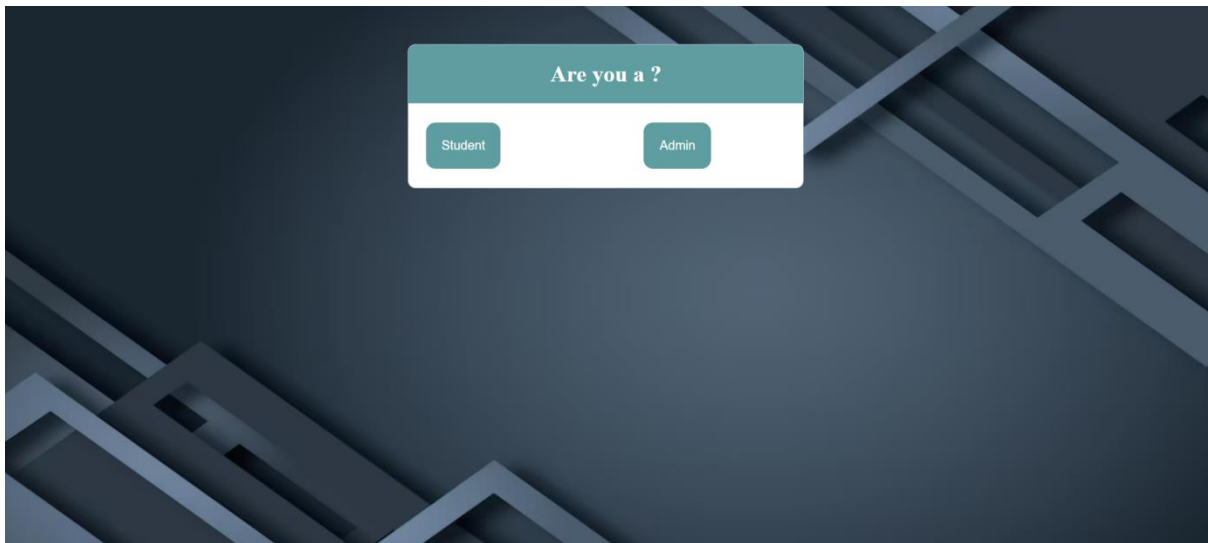
Future Directions and Recommendations:

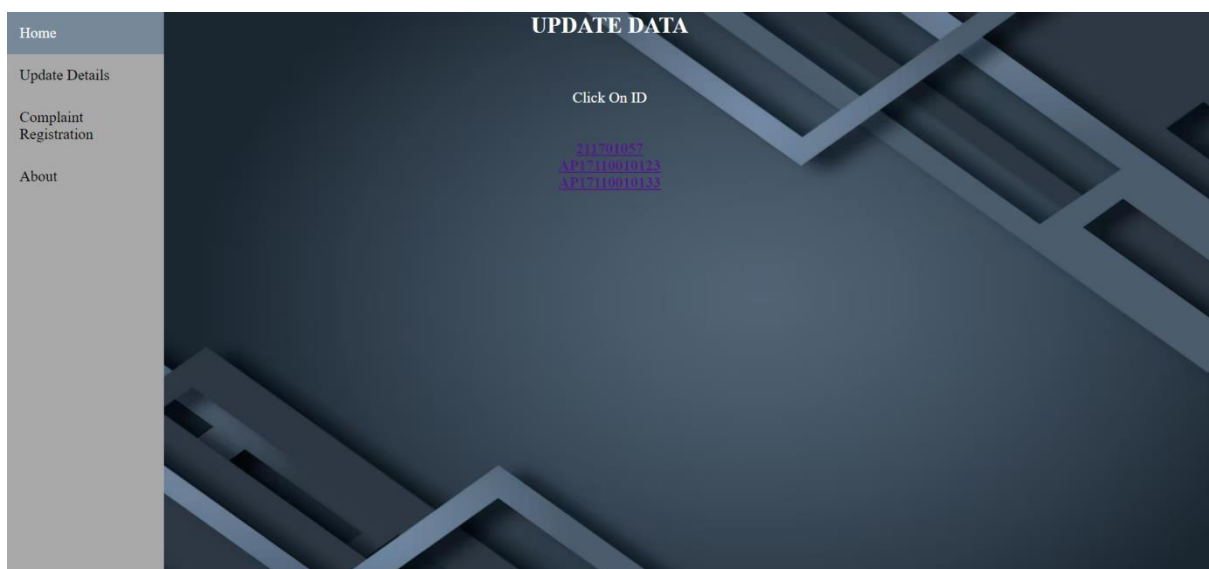
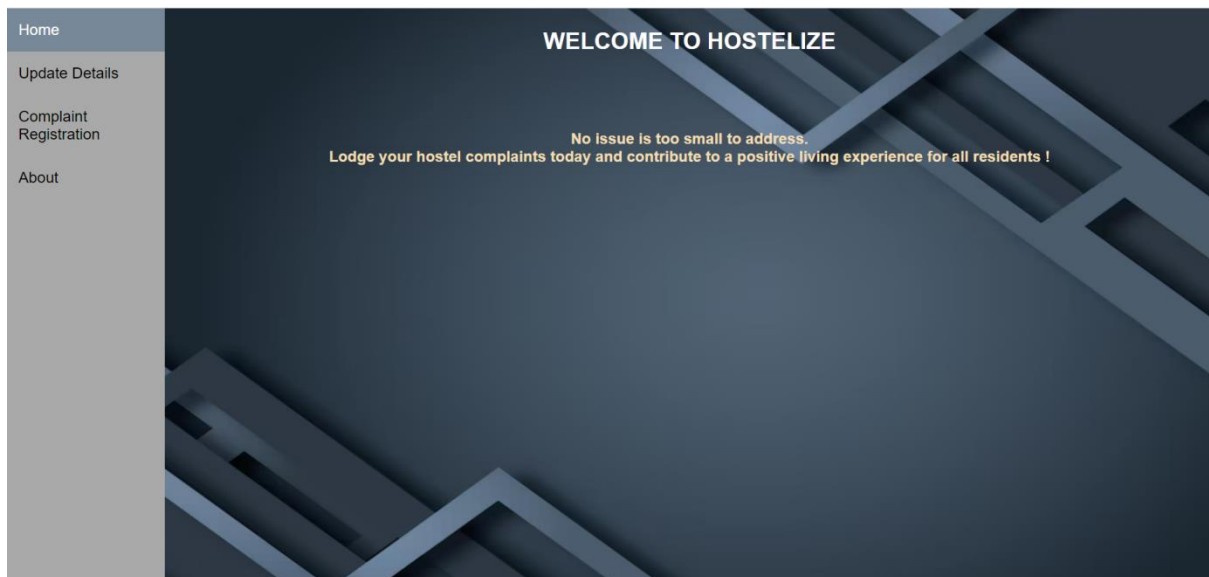
Propose future enhancements or features to further improve the website's functionality and user experience. This could include integration with additional modules for feedback collection, automated notifications, or analytics dashboards.

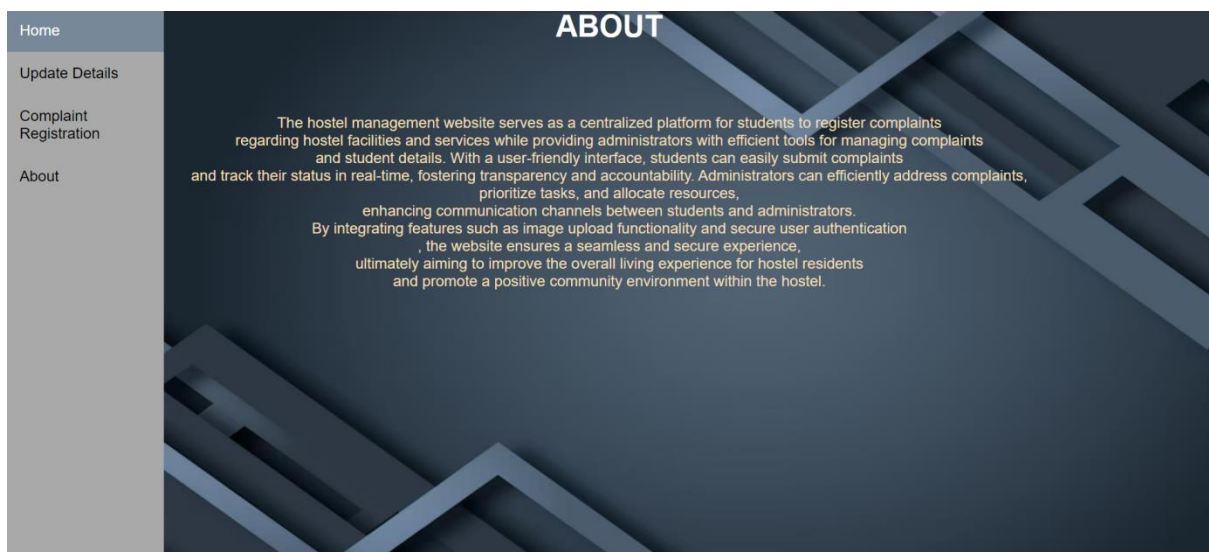
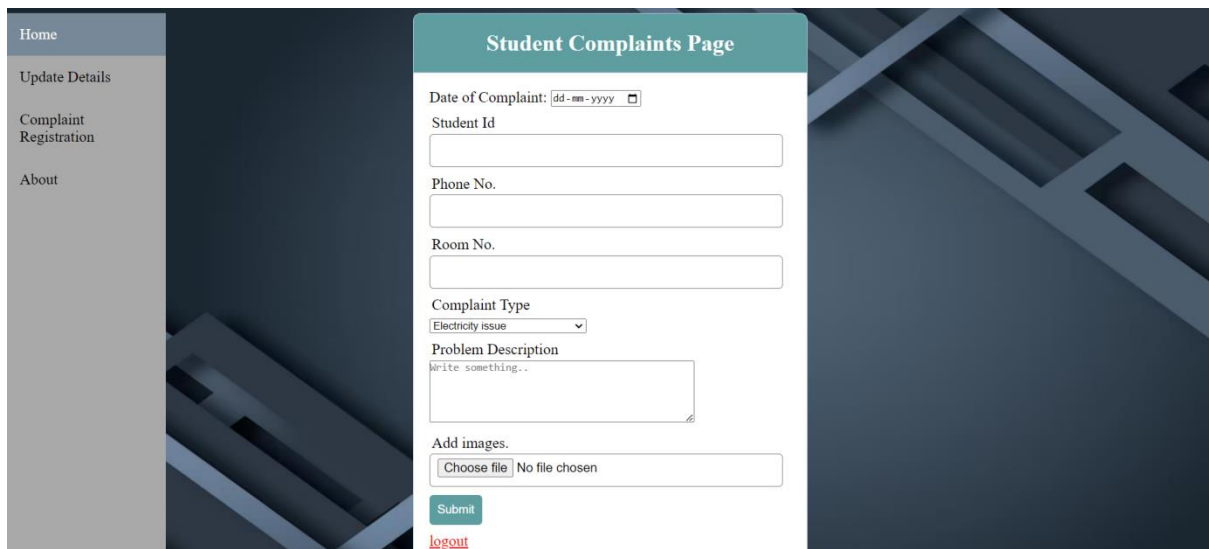
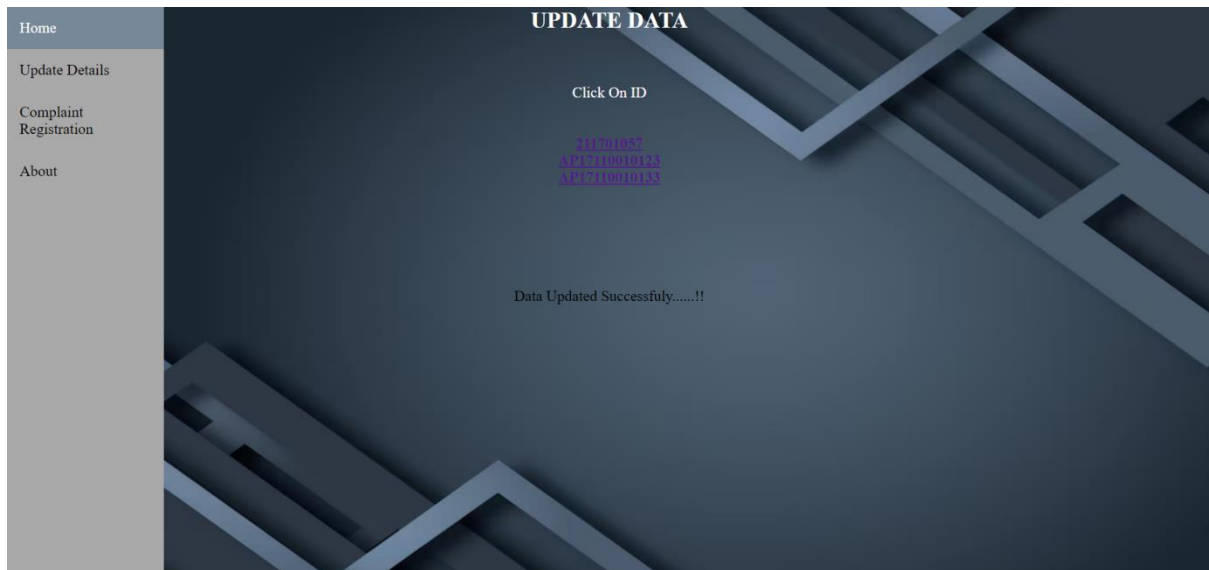
Provide recommendations for addressing any identified challenges or limitations in the current system, such as scalability issues, security concerns, or user adoption barriers.

Discuss potential opportunities for collaboration with other stakeholders, such as hostel management committees or student representatives, to continuously refine and enhance the complaint management process.

5.1 OUTPUT SCREENSHOTS:







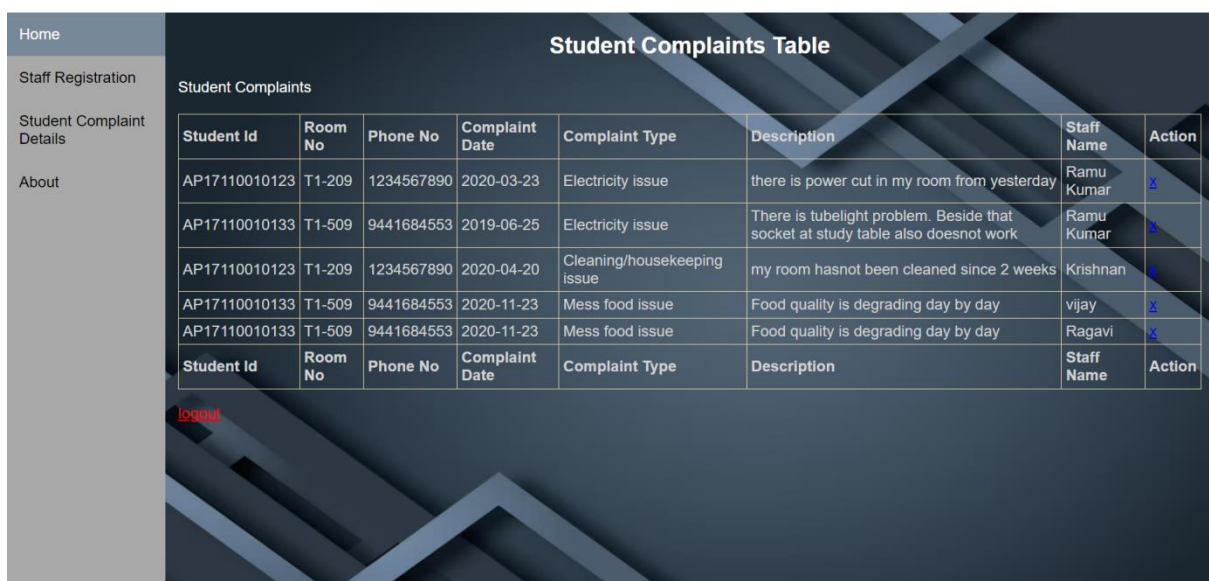
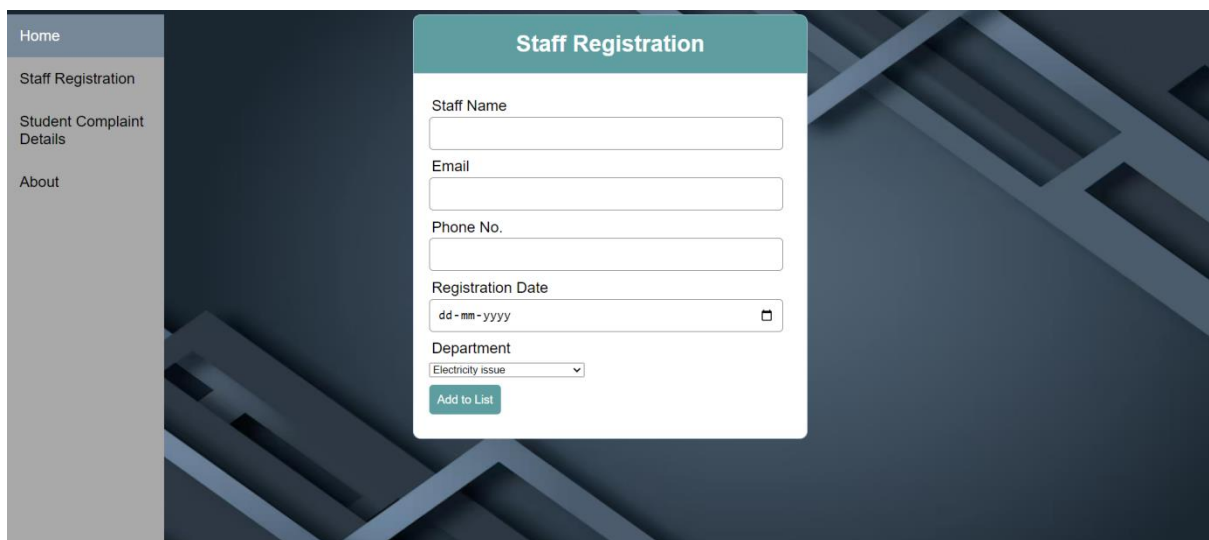
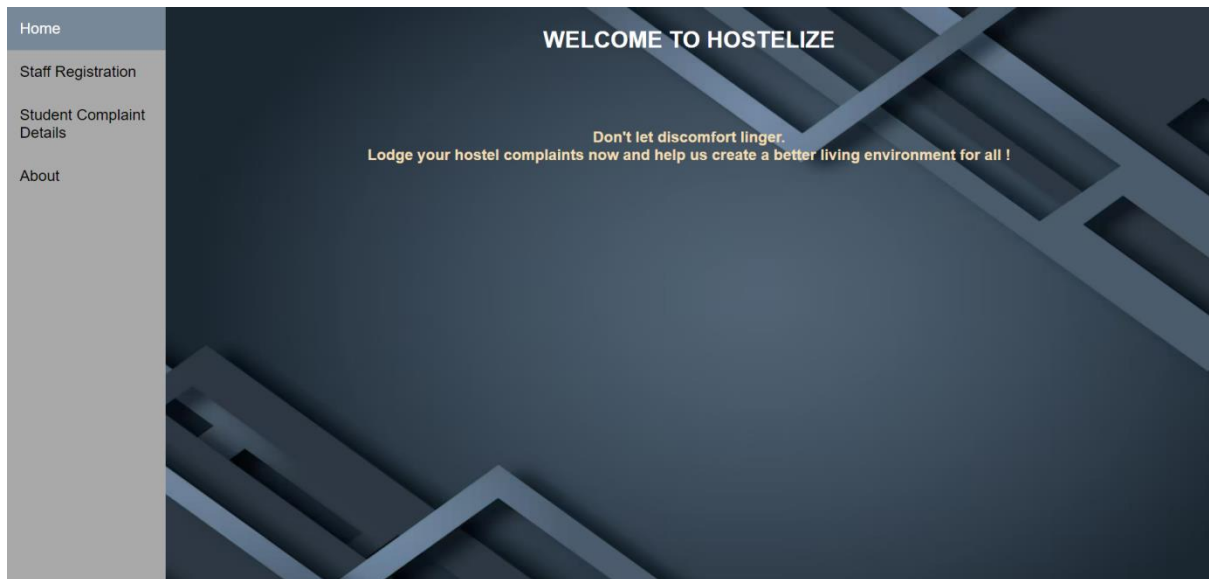


Fig 5.1 OUTPUT SCREENSHOTS

CHAPTER 6

CONCLUSION

The project to create a website for hostel complaint management has successfully enhanced communication, streamlined processes, and improved transparency. With efficient complaint registration and management functionalities, the website fosters better interaction between students and administrators. Despite encountering scalability challenges and user adoption barriers, valuable lessons have been learned. Moving forward, continued innovation and enhancements will be vital to further optimize the system's functionality, ensuring ongoing improvements in hostel management processes and student satisfaction.

The implementation of the hostel complaint management website has significantly improved the efficiency and effectiveness of handling student grievances. By providing a user-friendly interface for complaint registration and real-time tracking, the website has enhanced transparency and accountability within the hostel administration. Although certain technical constraints and scalability issues were encountered during development, the project has laid a strong foundation for future enhancements and optimizations. Continued iteration and refinement will be essential to maximize the website's impact on hostel management and student welfare.

REFERENCES

- [1] Title : Customer Complaint Management System Using SOA
Author : Jayawickrama,G.I.U
Published in:26-Jul-2021
<https://dl.ucsc.cmb.ac.lk/jspui/handle/123456789/4220>
- [2] Title: Student Complaint Management System
Author: Waleed Sadeq Noman Al-waeli , Norlida Binti Hassan
Published in :05-Jun-2022
<https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2440>
- [3] Title: Service Failure And Complaints Management In Higher Education
Institutions
Author: Steven kayambazinthu Msosa
Published in:03-Apr-2021
<https://ssbfnet.com/ojs/index.php/ijrbs/article/view/1098>
- [4] Title: Android Application For Complaint Management Systems
Author: Ms. D. Rejees Jenifa, Y. Savitha, K.Theebika
Published in:03-May-2023
<https://ijsrset.com/home/issue/view/article.php?id=IJSRSET2310232>