Noakhali Science & Technology University
Department of Computer Science & Telecommunication Engineering

A Project Report on

Design and Development of "Complaint, NSTU"

by

Mohammad Borhan Uddin
ID: ASH2101008M

Mohammad Billal Hossain
ID: MUH2101028M

Supervised by
Sharmin Akter Milu
Lecturer
Department of Computer Science & Telecommunication Engineering

Date : 27th January, 2025

DECLARATION

We hereby declare that this project has been completed by us under the supervision of Sharmin Akter Milu, Lecturer, Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University. We also affirm that neither this project nor any part of it has been submitted elsewhere for the award of any degree.

Mohammad Borhan Uddin
ID: ASH2101008M

Mohammad Billal Hossain
ID: MUH2101028M

Abstract

Efficiently handling complaints within an institution is crucial for improving the resolution process and enhancing overall productivity. This project, "Complaint, NSTU", focuses on developing a Complaint Management Application designed to streamline the submission and resolution of complaints. The application allows students to easily lodge complaints, while administrators or authorities can review, track, and resolve them through a structured process.

The project has been implemented using ReactJS for a user-friendly and interactive interface, and Node.js with Express.js for the back-end. MySQL has been used as the database management system to ensure secure and efficient data handling.

This system is expected to improve institutional efficiency, foster better communication, and contribute to a more responsive and productive environment.

Acknowledgment

We, the members of this project team, would like to express our sincere gratitude to all those who have contributed to the successful completion of this endeavor.

First and foremost, we extend our deepest appreciation to our esteemed lecturer, Mrs. Sharmin Akter Milu, for her unwavering guidance and mentorship throughout this project. Her insightful feedback, constant encouragement, and expert direction were instrumental in shaping the direction and quality of our work.

We also acknowledge the invaluable contributions of our team members, Mohammad Borhan Uddin (ASH2101008M) and Mohammad Billal Hossain (MUH2101028M). The dedication, collaborative spirit, and collective effort of the team were key to overcoming the challenges encountered during the development process.

Additionally, we would like to extend our heartfelt thanks to the NSTU Administration and the Department of Computer Science and Telecommunication Engineering (CSTE) for providing the necessary resources, including lab support, and for ensuring the availability of the ACM lab 24/7, which greatly contributed to the successful completion of this project. Their cooperation in facilitating our work was vital in making this project a reality.

Finally, we express our gratitude to our families and friends for their constant encouragement, understanding, and moral support. Their patience and motivation have been a source of strength and inspiration throughout the course of this project. Above all, we are profoundly grateful to Almighty Allah for His countless blessings, guidance, and support, which have enabled us to complete this project successfully.

Contents

1 Introduction

1.1 Introduction and Motivation for Complaint, NSTU

Efficient communication and timely resolution of issues are crucial for maintaining a productive and harmonious environment within any institution. At Noakhali Science and Technology University (NSTU), students often face challenges in lodging complaints due to the absence of a centralized and structured complaint management system. Traditional methods, such as verbal or written complaints, are not only time-consuming but also prone to inefficiencies and delays. These limitations create a communication gap between students and the administration, often leading to unresolved issues and dissatisfaction.

Complaint, NSTU aims to address these challenges by providing a centralized, web-based platform that allows students to submit complaints, track their resolution status, and interact with the administration efficiently. This system is designed to streamline complaint handling, ensure transparency, and enhance communication between students and the university administration

1.2 Objectives

To address these challenges, this project, **Complaint, NSTU**, introduces a digital solution for managing complaints efficiently. The primary objectives of this project are:

1. To provide a user-friendly platform for students to submit complaints.
2. To enable administrators to manage, track, and resolve complaints systematically.
3. To ensure transparency and improve communication between students and university authorities.
4. To enhance the efficiency of complaint resolution processes and foster a more responsive environment.

2 Literature Review

Effective complaint management systems have become an integral part of institutional administration to ensure transparency, efficiency, and user satisfaction. Numerous studies and existing systems have demonstrated the value of digital platforms in addressing user grievances effectively. However, many of these systems are either too generalized or lack specific features tailored for university environments like NSTU.

Several institutions rely on manual processes or basic web forms to manage complaints. For example, traditional paper-based systems and verbal communication are still prevalent in some universities, leading to inefficiencies such as lost complaints, delayed resolutions, and lack of transparency. These limitations have been highlighted in research on institutional grievance redress mechanisms, emphasizing the need for digital transformation.

2.1 Existing System

1. Generic Complaint Management Software

Many commercially available platforms, such as Zendesk and Freshdesk, are designed for large-scale organizations. While they offer features like ticketing systems and automated workflows, they are not tailored to address the unique needs of educational institutions, such as managing student-specific complaints or categorizing issues based on academic departments.

2. University-Specific Systems

Some universities have implemented custom-built solutions, such as online grievance portals or mobile apps. These systems often include features like complaint submission, tracking, and resolution but may lack user-friendly interfaces or real-time feedback mechanisms.

3. Research on Complaint Management

Studies in the field of complaint management, such as those by researchers in organizational behavior and system design, highlight critical components of effective systems:

- Easy accessibility and usability for all stakeholders.
- Transparent communication to build trust between users and administrators.
- Secure data handling to protect sensitive user information.
- Real-time tracking and notifications for better user engagement.

2.2 Proposed System

The **Complaint, NSTU** system addresses the shortcomings identified in existing systems by offering a platform tailored specifically for NSTU. Key differentiators include:

- A student-centric design that simplifies complaint submission and tracking.
- Role-based access for administrators, ensuring structured workflows.
- Integration of real-time feedback and commenting features to foster collaboration.
- The use of modern web technologies (ReactJS, Node.js, and MySQL) to ensure scalability, security, and efficiency.
- Adaptation to the institutional structure of NSTU, including departmental filtering and batch-based complaint categorization.

By bridging the gaps in existing solutions and incorporating feedback from stakeholders, **Complaint, NSTU** serves as a robust and comprehensive system for grievance redressal in a university context.

3 Methodology and Database Design

3.1 Definition

Methodology refers to a structured set of principles, methods, and processes used to guide the development and execution of a project or research.

3.2 Project Selection and Planning

We identified the need for an efficient complaint management system to streamline the complaint submission and resolution process for students and administrators. We assessed the feasibility of the project and defined its scope, ensuring it addressed the specific requirements of the users.

3.3 Requirement Gathering and Analysis

The first phase of the project involved gathering detailed requirements from key stakeholders. These stakeholders included students, faculty, and admin staff at NSTU (Noakhali Science and Technology University).

- Surveys and interviews were conducted to understand the users' needs, pain points, and desired features for the complaint submission system.
- The main functional requirements included user authentication, complaint submission, viewing complaint statuses, and admin functionality for approval or rejection.
- Non-functional requirements, such as system security, scalability, and responsiveness, were also considered.
- Educational email verification was established as a prerequisite for registration to ensure only legitimate students and staff could use the platform.

3.4 Use Case Modeling

A use case model describes different types of user interactions and their activities within a system. It also outlines the actions performed by the users. The key elements in a use case model include an actor, an event, and a use case. The major component is the actor.

In my project, *NSTU Complain*, there are two main actors:

- **Admin**

- **Student**

The admin is responsible for managing complaints submitted by students, while the students can submit complaints and track their status.

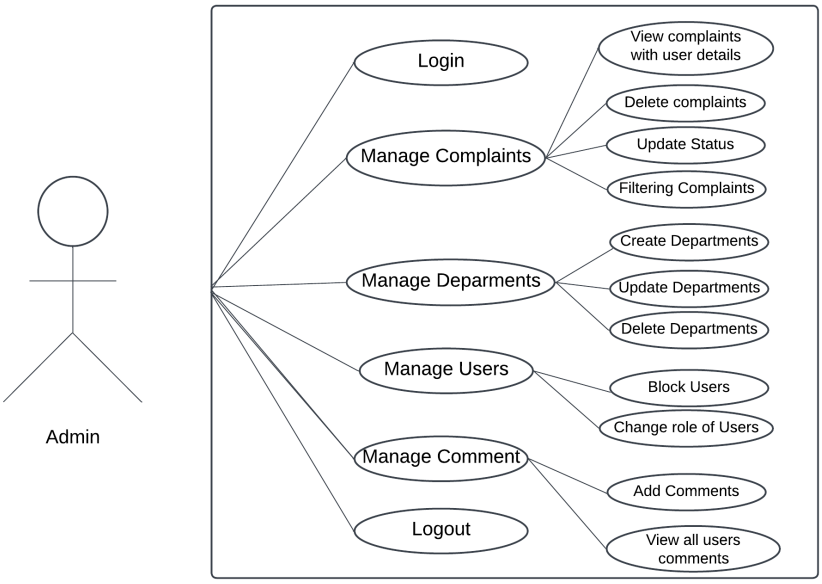

Figure 1: Use Case Diagram of Admin

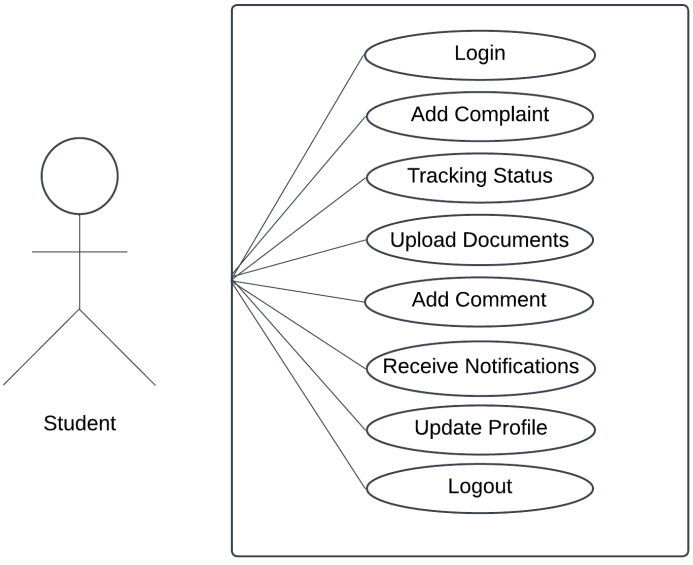

Figure 2: Use Case Diagram of Student

3.5 Database Design and Development

We designed the relational database structure, creating tables and relationships to store complaint data, user information, and complaint status. The database was developed to ensure data integrity and scalability, effectively handling multiple complaints and users.

3.5.1 Database Table Description

Departments Table

- **d_id**: VARCHAR(3) - Unique department ID
- **en_name**: VARCHAR(100) - Department name
- **fr_name**: VARCHAR(200) - Department name in French
- **est**: VARCHAR(20) - Establishment date
- **shortform**: VARCHAR(20) - Department abbreviation

Users Table

- **uid**: VARCHAR(100) - Unique user ID
- **name**: VARCHAR(100) - User name
- **roll**: VARCHAR(15) - User role
- **email**: VARCHAR(100) - User email
- **mobile**: VARCHAR(20) - User mobile
- **d_id**: VARCHAR(5) - Department ID
- **batch**: VARCHAR(5) - User batch
- **block**: TINYINT - User block

Complaints Table

- **c_id**: INT - Unique complaint ID
- **uid**: INT - User who filed complaint
- **details**: VARCHAR(167215) - Complaint details
- **created_at**: DATETIME - Complaint creation timestamp

- **updated_at**: DATETIME - Complaint last updated
- **status**: VARCHAR(20) - Complaint status

Comments Table

- **com_id**: INT - Unique comment ID
- **uid**: VARCHAR(100) - User who made comment
- **comment**: VARCHAR(65535) - Comment text
- **created_at**: VARCHAR(100) - Comment creation timestamp

Notifications Table

- **n_id**: INT - Unique notification ID
- **uid**: VARCHAR(100) - User for notification
- **message**: VARCHAR(65535) - Notification message
- **date**: VARCHAR(100) - Notification timestamp
- **_c_id**: INT - Related complaint ID

3.5.2 ER Diagram

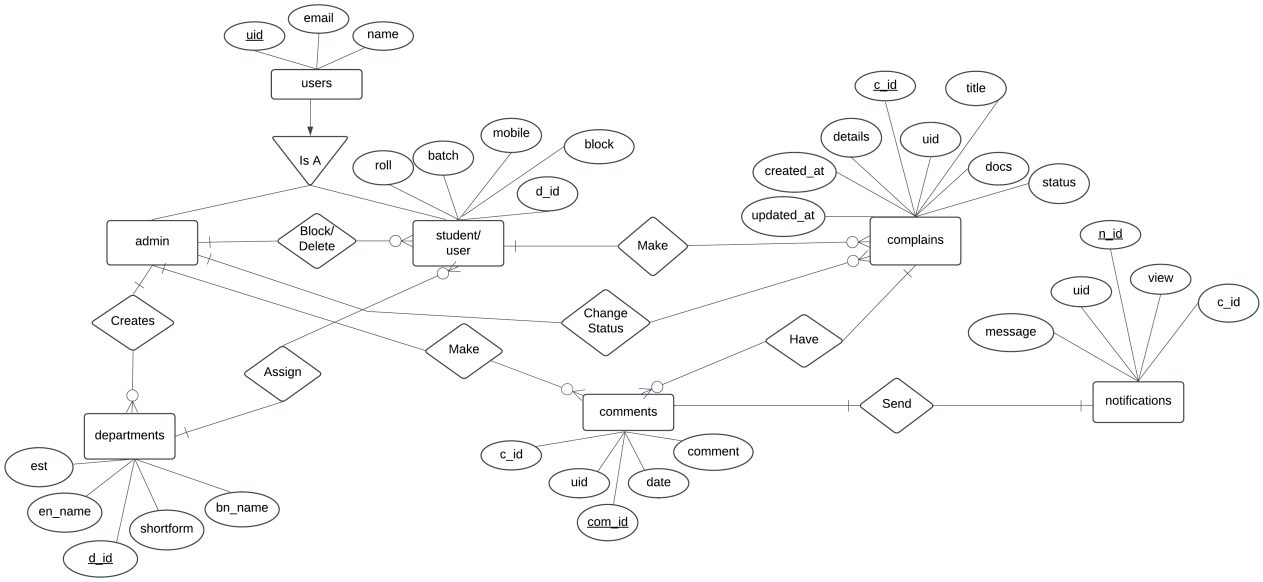

Figure 3: ER Diagram

3.5.3 Design of Tables

departments		
PK	d_id	varchar(3)
	en_name	varchar(100)
	bn_name	varchar(200)
	est	int
	shortform	varchar(20)

users		
PK	uid	varchar(100)
	name	varchar(100)
	roll	varchar(15)
	email	varchar(100)
	mobile	varchar(20)
	role	enum(5)
	d_id	varchar(5)
	batch	varchar(5)
	block	tinyint

complains		
PK	c_id	int
FK	uid	varchar(100)
	details	mediumtext(16777215)
	docs	text(65535)
	created_at	datetime
	updated_at	datetime
	status	varchar(20)
	title	varchar(300)

comments		
PK	com_id	int
	uid	varchar(100)
	date	varchar(100)
	comment	text(65535)
FK	c_id	int

notifications		
PK	n_id	int
	uid	varchar(100)
	view	tinyint
	message	varchar(300)
	date	varchar(100)
FK	c_id	int

Figure 4: Design of Table

3.5.4 Relationships of tables

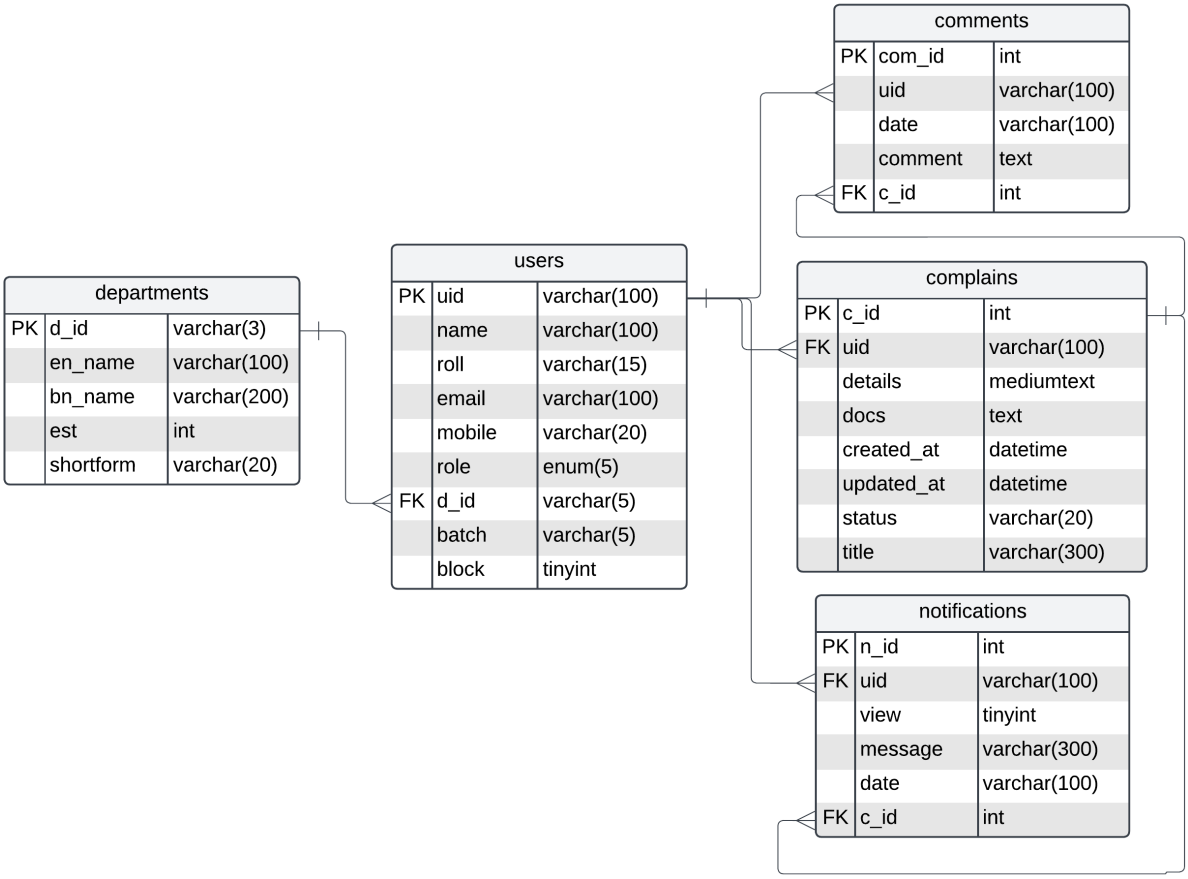

Figure 5: Relationship of tables

3.6 Technology and Programming Language Selection

Selecting the right technologies is crucial for ensuring system scalability, performance, and user experience. In this project, we chose tools that align with the project's goals, providing an efficient, maintainable, and user-friendly solution.

3.6.1 Frontend

We selected **ReactJS** for front-end development due to its component-based architecture, which allows for efficient and reusable UI components. To ensure a responsive and visually appealing user interface, we used **Material UI (MUI)** for pre-built UI components, and **Tailwind CSS** for flexible styling and layout

management. Additionally, we incorporated **Helmet** for managing the document head, ensuring proper SEO and title management for each page. ReactJS combined with these tools provides a seamless, user-friendly experience for both students and administrators.

3.6.2 Backend

For the backend, we chose **Node.js** with **Express** due to their scalability, speed, and asynchronous nature. Node.js enables us to build a highly efficient and scalable application with a non-blocking I/O model, while Express simplifies routing and middleware integration. **MySQL2** was used as the database client for connecting Node.js to the MySQL database, allowing for seamless database queries and operations.

3.6.3 Databases

We selected MySQL as the relational database for storing complaint data due to its reliability, performance, and ability to handle structured data with relationships. MySQL ensures efficient management of user and complaint records, maintaining data consistency and scalability as the system grows.

3.7 Hardware and Software Requirements

The Requirement Analysis phase is a critical part of software development, where the technical needs for hardware, software, and tools are identified and defined.

3.7.1 Hardware Requirements

- Processor : Intel(R) core(TM) i5-5005 U CPU @ 2.00 GHZ 2.00GHZ
- RAM : 4GB
- Hard disk : 1TB
- Monitor : VGA/SVGA

3.7.2 Software Requirements

- Operating System : Windows 10
- IDE : Visual Studio Code
- Browser : Chrome, Edge

- Version Control : Git and GitHub
- Package Manager : NPM
- Database : MySql

3.8 System Implementation, Development, and Testing

The implementation phase involved translating design specifications into functional code, using suitable programming languages and frameworks. Comprehensive testing was conducted throughout the development lifecycle to ensure that the system met functional requirements and performed reliably under various condition

3.9 Deployment and Final Presentation

The completed system was deployed to a cloud server for production use. After deployment, we presented the system to the stakeholders, demonstrating its features and collecting feedback for future improvements. Additionally, a comprehensive project report was prepared, documenting the system's design, development process, and key functionalities, providing stakeholders with a detailed overview of the project.

4 Authentication and Authorization

4.1 Definition

The Authentication and Authorization Mechanism verifies user identities and controls access based on roles. It uses Firebase Authentication for authentication and implements role-based access control (RBAC) to manage permissions for different user types.

4.2 Authentication Mechanism

4.2.1 User Signup and Login

Users log in through the 'Sign Up with Google' feature, utilizing Firebase Authentication, which eliminates the need for password creation or management. Firebase Authentication handles the validation process and retrieves details from the user's Google account, including the Full Name and the Email Address. The Full Name is provided directly by Firebase Authentication, while the Email Address follows the format provided by the user's Google account.

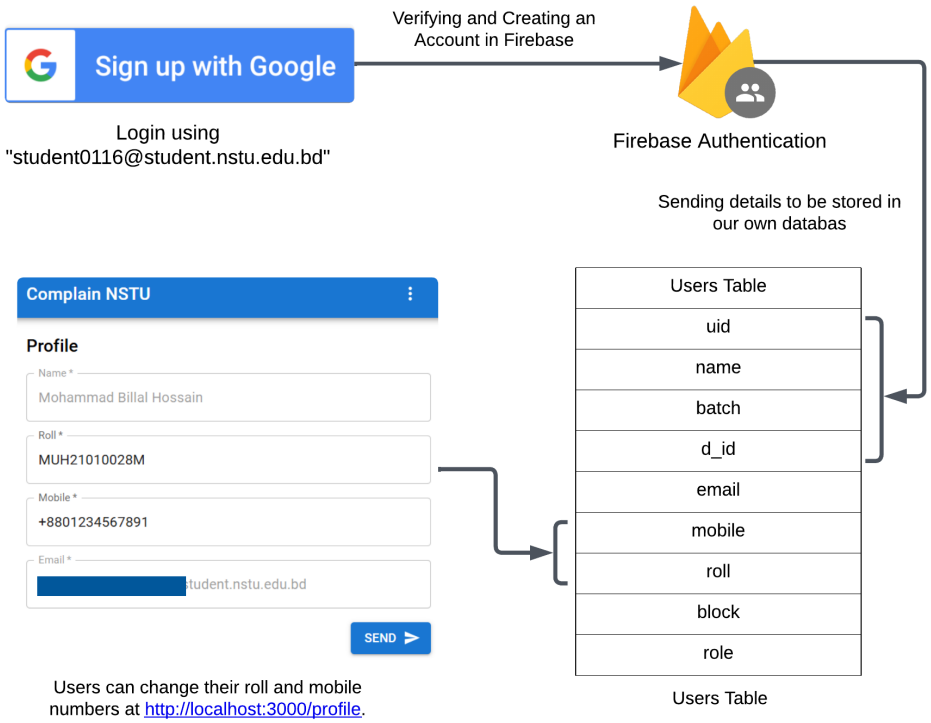

Figure 6: Authentication Mechanism

4.2.2 Database Integration

After successful authentication, the user's email address is verified to ensure it follows the structure: studentnameDepartmentIdBatch@student.nstu.edu.bd. If the email matches the expected format, additional information is extracted, including the Department, derived from the DepartmentId section, and the Batch, parsed from the Batch segment. These details, along with the user's role (Student or Admin) and Firebase User ID, are then stored in the database.

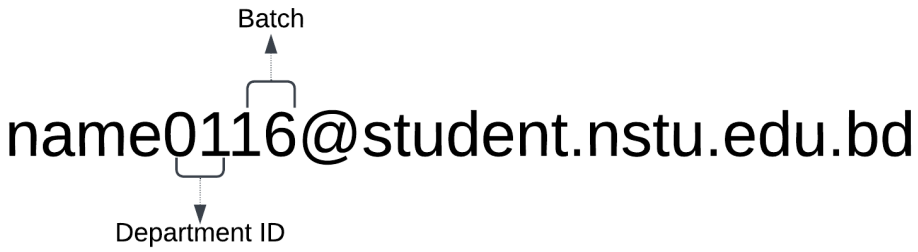

Figure 7: Identifying the Department and Batch

4.3 Authorization Mechanism

A Role-Based Access Control (RBAC) system is implemented to restrict access based on user roles. The Admin role has full access to manage complaints and block users, while the Student role has limited access, allowing them to submit and track their own complaints. Middleware functions are used to validate the user's session token, email, and role before granting access to protected resources, ensuring proper authorization for each user.

5 Software Design

5.1 Definition

Software design is an iterative process that translates requirements into a blueprint for building software. Initially, the design provides a high-level view, aligning with the system's objectives and detailed data, functional, and behavioral requirements.

5.2 A Closer Look at Our Web App

Login

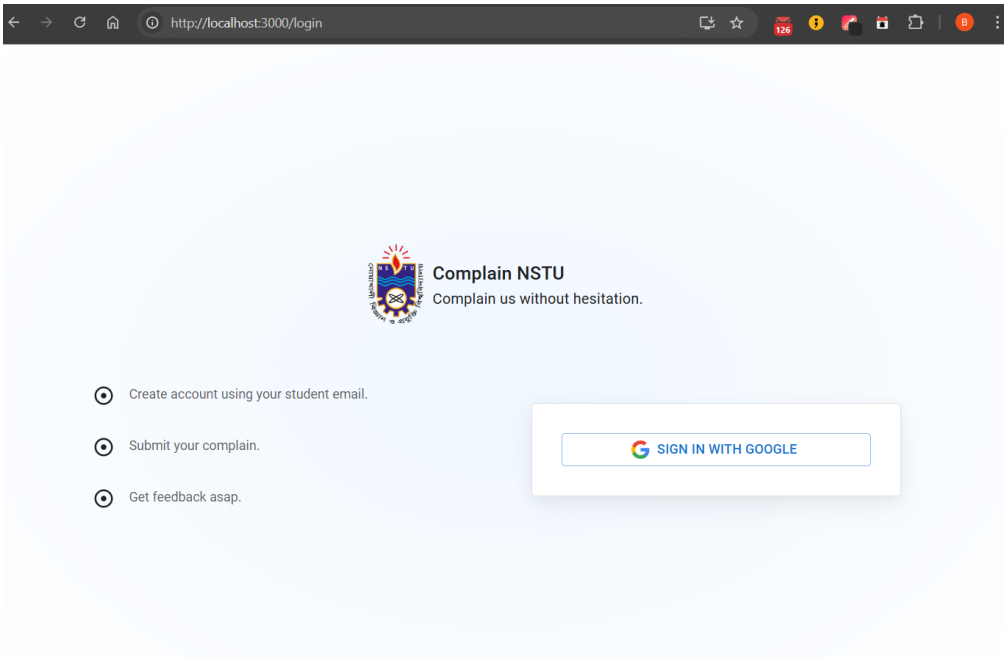

Figure 8: Login Page

Student

Home

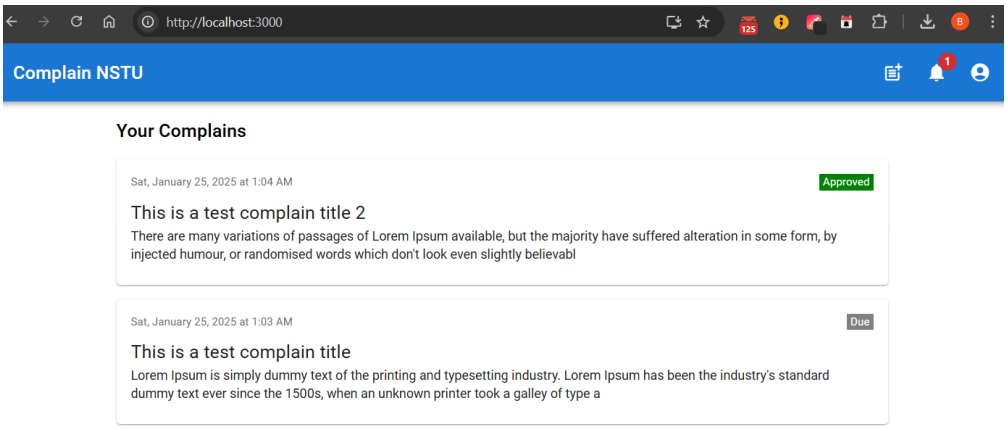

Figure 9: Home Page

Add Complaint

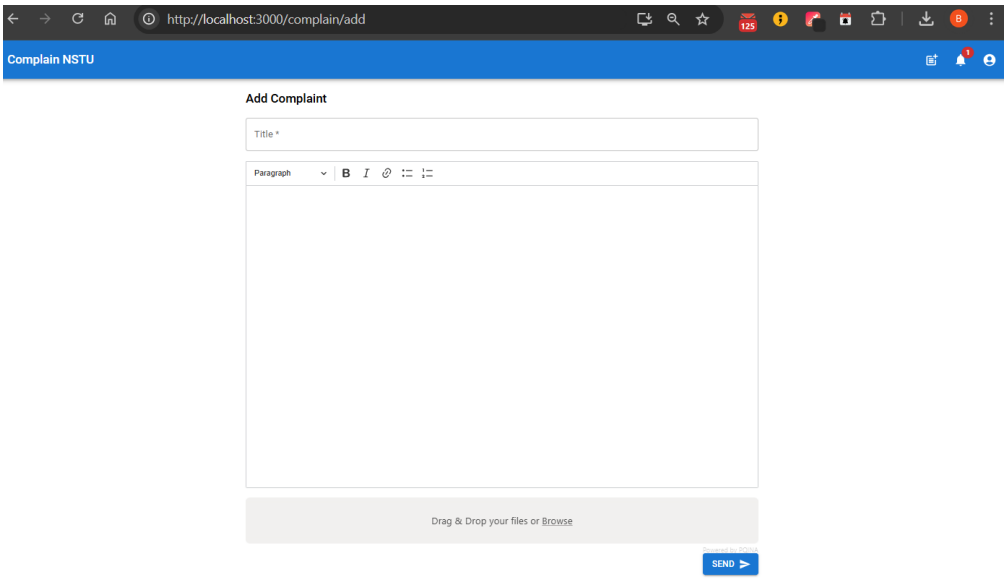

Figure 10: Add Complaint Page

Complain Details

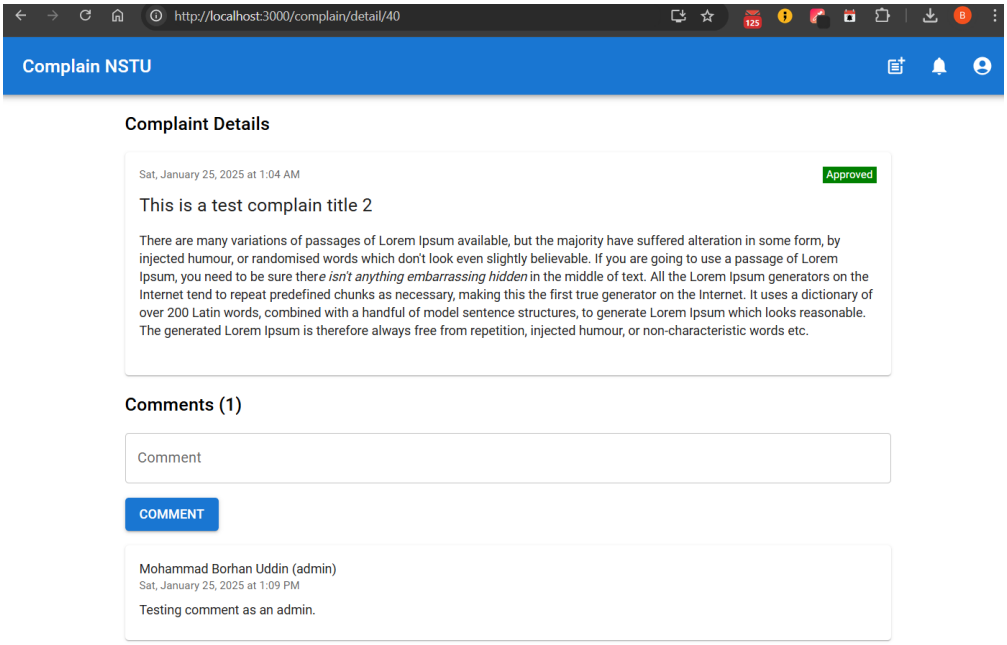

Figure 11: Complain Details Page

Notifications

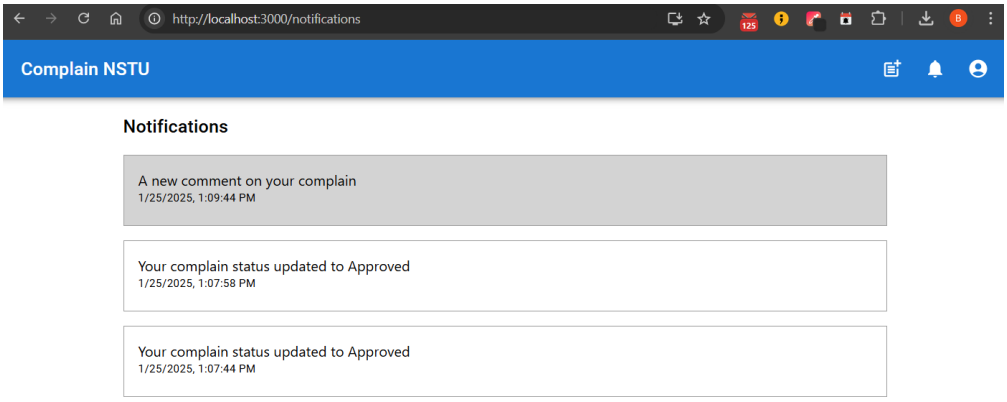

Figure 12: Notification

Admin Panel

Home

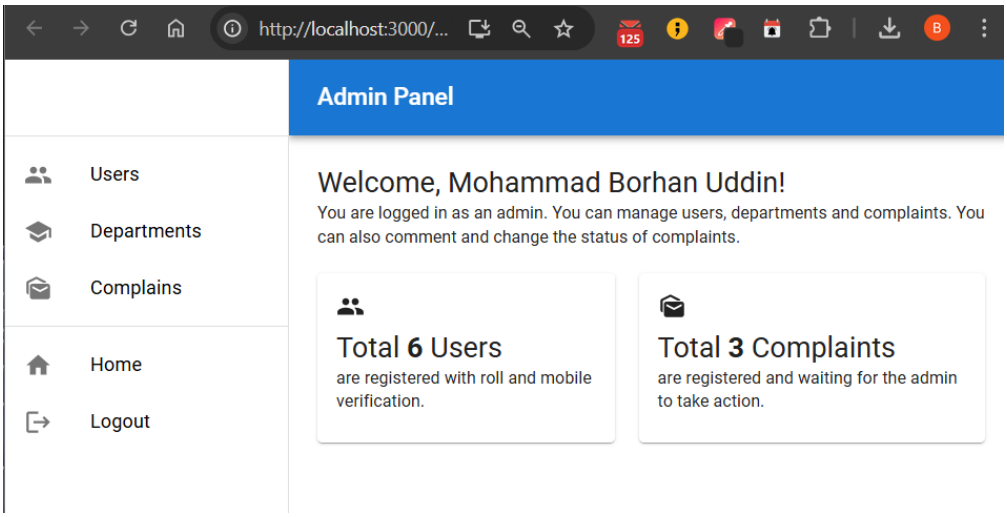

Figure 13: Admin Page

Department

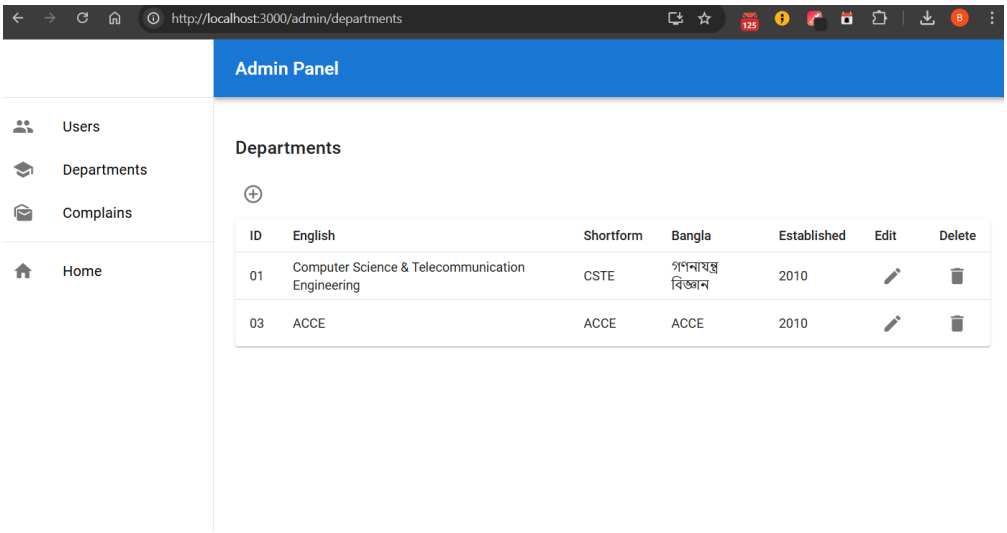

Figure 14: Department

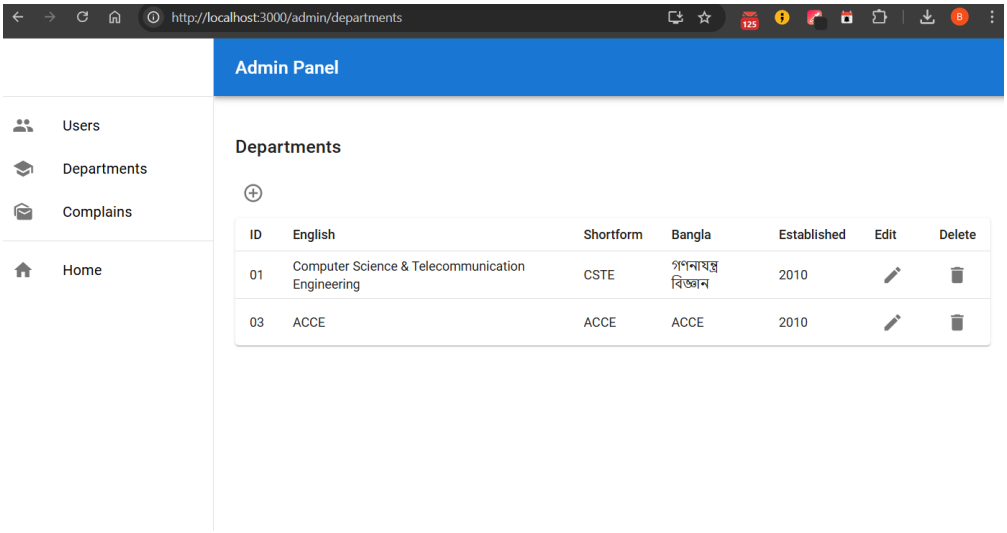

Figure 15: Creating new Department

Complain

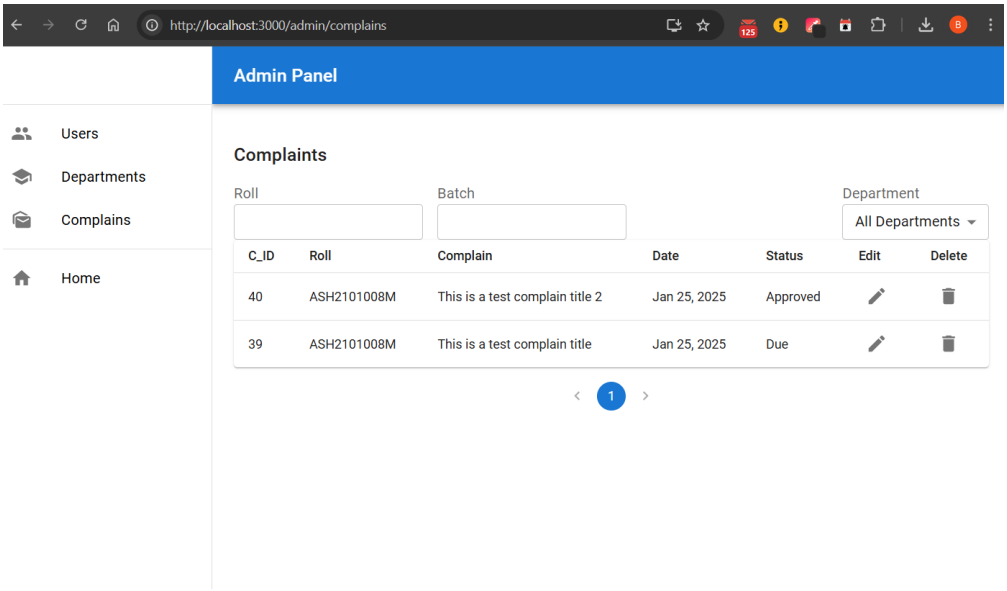

Figure 16: Complaints View

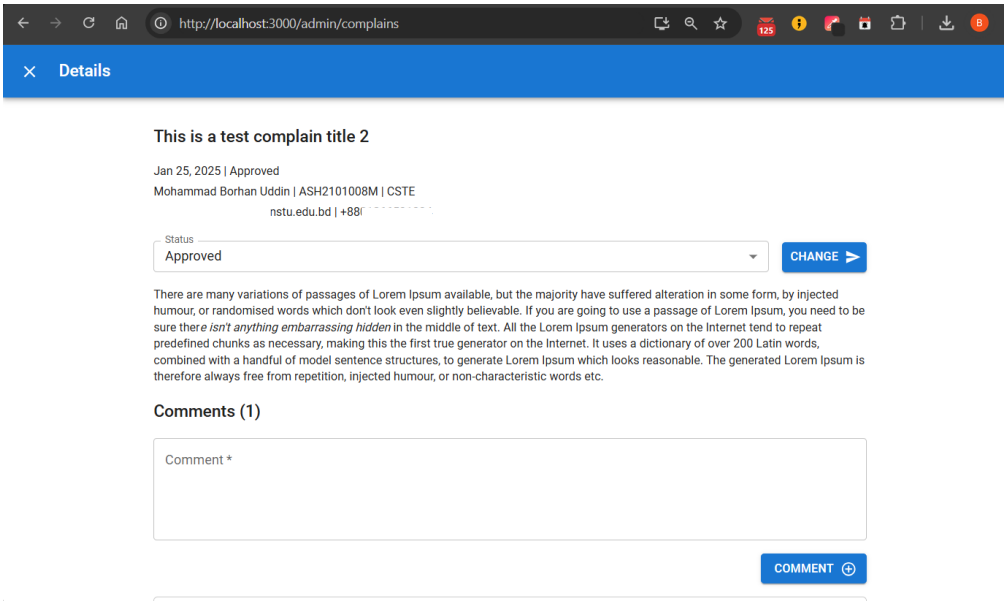

Figure 17: Complaints Managing

User

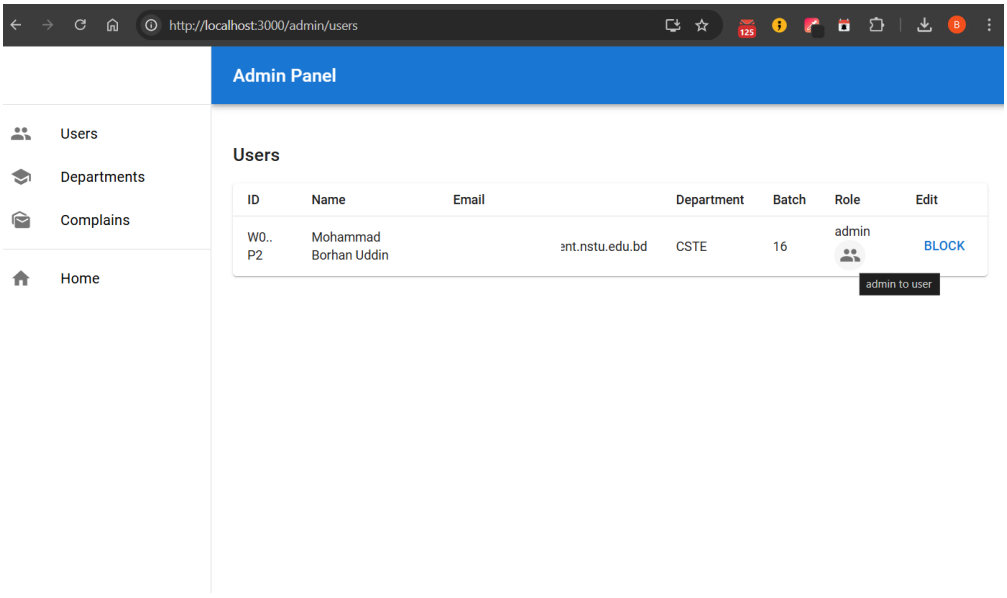

Figure 18: User Management

5.3 Github Repository

https://github.com/borhan008/complain_nstu

6 Conclusion, and Recommendations

6.1 Conclusion

In conclusion, the **Complaint, NSTU** web application serves as a convenient and efficient platform for students to submit complaints and for admins to manage them. The use of Google Sign-In simplifies the authentication process and ensures that user data is securely handled. However, the system still has several limitations, such as the lack of an admin-specific login pattern, limited user roles, and basic search features. Future improvements could include the introduction of a dedicated mobile app, enhanced admin access control, advanced filtering and search capabilities, and the support of additional user roles. Despite these limitations, the project meets the fundamental requirements and provides a good foundation for further enhancements.

6.2 Recommendations

The software is designed for general use with ease. However, if any modifications or additional features are required, users are encouraged to provide suggestions.

- **Expand User Roles**: To accommodate future use cases, consider introducing more user roles such as Moderator, Guest, or Support, to provide more granular access control.
- **Implement Custom Authentication**: Evaluate the possibility of integrating other authentication mechanisms or services to provide more flexibility and control over the authentication process.
- **Optimize Scalability**: To address scalability concerns, implement performance optimizations such as database indexing, caching, or cloud services to better handle growing data and user load.
- **Enhance Security Measures**: Ensure that security best practices are followed by implementing features like multi-factor authentication (MFA) and regular security audits.

7 References

1. Firebase. (2023). *Firebase Authentication*. Retrieved from `https://firebase.google.com/docs/auth`
2. Google Developers. (2023). *Google Sign-In for Websites*. Retrieved from `https://developers.google.com/identity/sign-in/web/sign-in`
3. Mozilla Developer Network. (2023). *Web Storage API*. Retrieved from `https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage`
4. W3C. (2018). *Web Application Security*. Retrieved from `https://www.w3.org/Security`
5. MDN Web Docs. (2023). *JavaScript Functions*. Retrieved from `https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions`
6. Node.js Foundation. (2023). *Node.js Documentation*. Retrieved from `https://nodejs.org/en/docs/`
7. MySQL Documentation. (2023). *MySQL Reference Manual*. Retrieved from `https://dev.mysql.com/doc/`
8. ReactJS. (2023). *React - A JavaScript Library for Building User Interfaces*. Retrieved from `https://reactjs.org/`
9. Material-UI. (2023). *Material UI Documentation*. Retrieved from `https://mui.com/`
10. Ahmad, M., & Raza, S. (2020). *Building Scalable Web Applications: A Practical Approach*. Journal of Web Development, 15(2), 45-56.
11. Smith, R., Doe, J., & Brown, K. (2019). *User Authentication in Web Applications*. In *Proceedings of the International Conference on Web Security* (pp. 35-41). Springer.