***The World Islamic Sciences and Education University***

جامعة العلوم الاسلامية العالمية

Faculty of Information Technology

كلية تكنولوجيا المعلومات

****

GRADUATION PROJECT

**Title**

*Wise Complaint Hub*

**Students**

Abdalrahaman Mohammad Ahmad Abodaeyh CS 3210601001

Ahmad Haitham Hasan Al Semren CS 3220601094

Mahmoud Mohammad Mahmoud Al Khawaja SE 3210605012

**Supervisor**

**Dr. Noor Al-Qudah**

SEMESTER II

2024/2025

**Acknowledgements**

We would like to sincerely thank The World Islamic Sciences and Education University for providing a nurturing and enriching academic environment that fostered both personal and intellectual growth.

This project would not have been possible without the unwavering support and invaluable contributions of many individuals who played a vital role in its successful completion.

First and foremost, we extend our deepest gratitude to our esteemed supervisor, Dr. Noor Al-Qudah, for her expert guidance, constructive feedback, and continuous encouragement. Her insightful advice and dedication were instrumental in shaping this project and bringing it to fruition. It has been a great honor to work under her mentorship.

We are also profoundly grateful to our families for their constant love, patience, and support. Their belief in us provided the strength and motivation needed to overcome challenges and stay focused throughout this journey.

Finally, we would like to express our heartfelt appreciation to everyone who, in one way or another, contributed their time, knowledge, and resources to support this project. Your generosity and assistance have been invaluable.

**Abstract**

This project presents a Complaint Management System designed specifically for university environments to streamline the process of submitting, tracking, and resolving complaints. The system provides an easy-to-use interface for students, faculty, and staff to file complaints efficiently while allowing administrators to manage and assign cases effectively. Built using ASP.NET Core with Entity Framework, the system ensures data integrity, security, and role-based access control. The implementation improves transparency and accountability within the university, enabling timely resolution of issues and enhancing overall satisfaction. Testing confirms the system’s reliability, scalability, and usability, making it a valuable tool for modern university administration.

Contents

[CHAPTER 1 1](#_Toc199848381)

[INTRODUCTION 1](#_Toc199848382)

[1.2 Problem Statement 2](#_Toc199848383)

[1.3 Project objectives 3](#_Toc199848384)

[1.4 Research strategy 3](#_Toc199848385)

[1.5 Scope 4](#_Toc199848386)

[1.6 Gant chart 4](#_Toc199848387)

[1.7 Project outline 5](#_Toc199848388)

[2.1 Overview 8](#_Toc199848389)

[2.3 Related Work 8](#_Toc199848390)

[2.3.1 Discussions 8](#_Toc199848391)

[2.3.2 Issues and challenges faced by the other project 9](#_Toc199848392)

[2.4 Summary 10](#_Toc199848393)

[3.1 Overview 12](#_Toc199848394)

[3.2 Feasibility Study 12](#_Toc199848395)

[3.3 Methodology 13](#_Toc199848396)

[3.3.1 Requirements 13](#_Toc199848397)

[3.4 Methodology Process 16](#_Toc199848398)

[CHAPTER 4 22](#_Toc199848399)

[DESIGN MODELS 22](#_Toc199848400)

[4.1 Overview 22](#_Toc199848401)

[4.2 Context diagram-0 23](#_Toc199848402)

[4.3 Data flow Diagram-1 23](#_Toc199848403)

[4.4 Use Case Diagram 24](#_Toc199848404)

[4.5 ER Diagram 25](#_Toc199848405)

[4.6 Relational Model 26](#_Toc199848406)

[5.1 Overview 28](#_Toc199848407)

[5.2 Testing methodologies 28](#_Toc199848408)

[5.2.1 Unit Testing Results 28](#_Toc199848409)

[5.2.1.1 Login: 28](#_Toc199848410)

[5.2.2 Integration Testing Results 33](#_Toc199848411)

[5.2.3 System Testing Results 33](#_Toc199848412)

[5.2.4 Acceptance System Results 34](#_Toc199848413)

[5.3 Deployment and Hosting 34](#_Toc199848414)

[5.4 Discussion and evaluation 35](#_Toc199848415)

[CHAPTER 6 36](#_Toc199848416)

[CONCLUSION AND FUTURE WORKS 36](#_Toc199848417)

[6.1 Overview 37](#_Toc199848418)

[6.2 Summary about the project 37](#_Toc199848419)

[6.3 Achieved objectives 37](#_Toc199848420)

[6.4 Main contributions of the work 38](#_Toc199848421)

[6.5 Limitation 38](#_Toc199848422)

[6.6 Future Work 38](#_Toc199848423)

# List of tables

[Table 1 System Comparison 10](#_Toc199849716)

[Table 2 Feasibility Study 12](#_Toc199849717)

[Table 3 Real cost for our project 12](#_Toc199849718)

[Table 4 Functional 14](#_Toc199849719)

[Table 5 Non-functional Requirements 15](#_Toc199849720)

[Table 6 Scrum Sprints 18](#_Toc199849721)

# List of figures

[Figure 1 Project Sprint 4](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849722)

[Figure 2 Gantt Chart 5](#_Toc199849723)

[Figure 3 Sprints Process 17](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849724)

[Figure 4 Context diagram- 23](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849725)

[Figure 5 Data flow Diagram 24](#_Toc199849726)

[Figure 6 Use Case Diagram 25](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849727)

[Figure 7 Relational model 26](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849728)

[Figure 8 Login Page Email & Password 28](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849729)

[Figure 9 Invalid Email or Password 29](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849730)

[Figure 10 Validation Please Enter A valid Email Address 29](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849731)

[Figure 11 must be at least 4 ch password 30](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849732)

[Figure 12 Email and Password Is Required 30](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849733)

[Figure 13 Complaint submission 31](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849734)

[Figure 14 Required Title, Description, Catogory 31](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849735)

[Figure 15 Make Assignee (SuperAdmin) 32](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849736)

[Figure 16 Make Action And Comment (Complaint Handler) 32](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849737)

[Figure 17 select assignee 33](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849738)

[Figure 18 asp.net hosting 34](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849739)

[Figure 19 dashboard super admin 41](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849740)

[Figure 20 dashboard Student 41](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849741)

[Figure 21 Manges role 42](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849742)

[Figure 22 Edit role and Claims 42](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849743)

[Figure 23 manage user 43](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849744)

[Figure 24 manage complaint type 43](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849745)

[Figure 25 edit email setting 44](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849746)

[Figure 26 add new complaint 44](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849747)

[Figure 27 add new announcement 45](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849748)

[Figure 28 assign role to user 45](file:///C:\Users\gassa\Downloads\قالب%20مشروع%20التخرج%20Last%20edit.docx#_Toc199849749)

**List of abbreviations**

|  |  |
| --- | --- |
| **WCH** | WISE Complaint Hub |
| **WISE** | The world Islamic Science & Education University |
| **IT** | Information Technology |
| **HTML** | Hypertext Markup Language |
| **CSS** | Cascading Style Sheets |
| **JS** | Java Script |
| **SQL** | Structured Query Language |
| **UI** | User Interface |
| **ER** | Entity Relationship |
| **CS** | Computer Science |
|  |

# CHAPTER 1

# INTRODUCTION

* 1. **Overview**

In modern university environments, efficiently managing and resolving student complaints is crucial for maintaining a positive academic atmosphere and improving institutional services. Traditional complaint handling methods often face challenges such as delays, lack of transparency, and difficulties in tracking the resolution status.

This project aims to develop a **Complaint Management System** specifically for **students** to submit, track, and resolve their complaints. By leveraging a user-friendly digital platform, the system ensures timely responses, transparent communication between students and university staff, and organized complaint records.

The system will facilitate improved accountability and satisfaction by enabling easy complaint submission exclusively by students, automated notifications, and detailed reporting. Ultimately, it seeks to enhance the overall student experience and institutional effectiveness through a secure, efficient, and accessible complaint management solution.

1.2 Problem Statement

In many universities, the process of handling student complaints is often inefficient and lacks transparency. Currently, students submit complaints through manual or fragmented channels such as paper forms or email, which leads to delays in processing and difficulty in tracking the status of their complaints.

This manual process is time-consuming and prone to errors, causing frustration among students and inefficiency for administrative staff. There is often no clear communication channel between students and university officials, resulting in complaints being lost or ignored.

Furthermore, the lack of a centralized system prevents effective reporting and analysis of complaint trends, which limits the university’s ability to improve its services and respond proactively to student needs.

Therefore, there is a need for a digital Complaint Management System that streamlines complaint submission, tracking, and resolution exclusively for students, ensuring timely handling, transparency, and accountability in addressing their concerns.

# 1.3 Project objectives

This project aims to develop an efficient and transparent Complaint Management System exclusively for university students. The main objectives are:

1. To provide students with a simple and accessible digital platform to submit their complaints easily.
2. To enable real-time tracking of complaint status, ensuring transparency throughout the resolution process.
3. To reduce the delays and errors associated with traditional manual complaint handling methods.
4. To maintain organized records and generate detailed reports for better analysis and continuous improvement of university services.
5. To ensure data security and privacy of all submitted complaints and users’ information.

# 1.4 Research strategy

This project follows a systematic approach to develop the Complaint Management System by integrating research, design, and development phases:

* Literature Review: Relevant academic papers, articles, and existing systems were reviewed to understand best practices and common challenges in complaint management.
* Requirements Gathering: Information was collected through surveys and interviews with students and university staff to identify key features and user needs.
* System Design: The system architecture, database schema, and user interface designs were created based on the gathered requirements.
* Development Methodology: An iterative development approach (e.g., Agile) was adopted to allow continuous feedback and improvement throughout the project lifecycle.
* Implementation: The system was developed using appropriate programming languages and tools suited to ensure security, usability, and performance.
* Testing and Evaluation: The system was tested with real users to identify and fix issues, and to assess functionality, usability, and effectiveness.

This framework ensures that the system meets user needs and operates efficiently within the university environment. (Pressman, R. S. (2014). *Software Engineering: A Practitioner’s Approach* (8th ed.). McGraw-Hill Education).

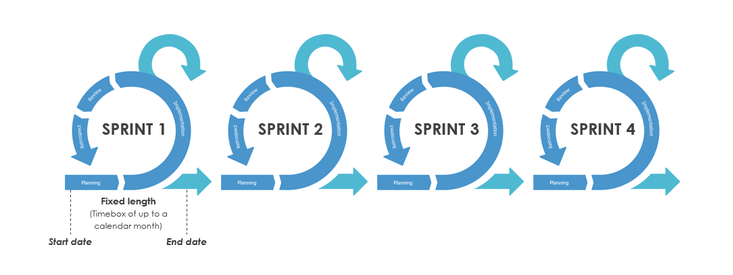


Figure Project Sprint

# 1.5 Scope

1. This project is designed specifically for the university to improve how complaints are managed within its community.
2. The system requires a stable internet connection to function properly.
3. It relies on the university’s servers, so there might be technical issues or downtime during busy periods.
4. The system does not cover complaints related to other institutions like government bodies or external schools.
5. Users (students, faculty members, and staff) need compatible devices (computers or smartphones) to access the system and submit complaints.
6. The system administrator must have full access to manage and monitor the complaints effectively.

# 1.6 Gant chart

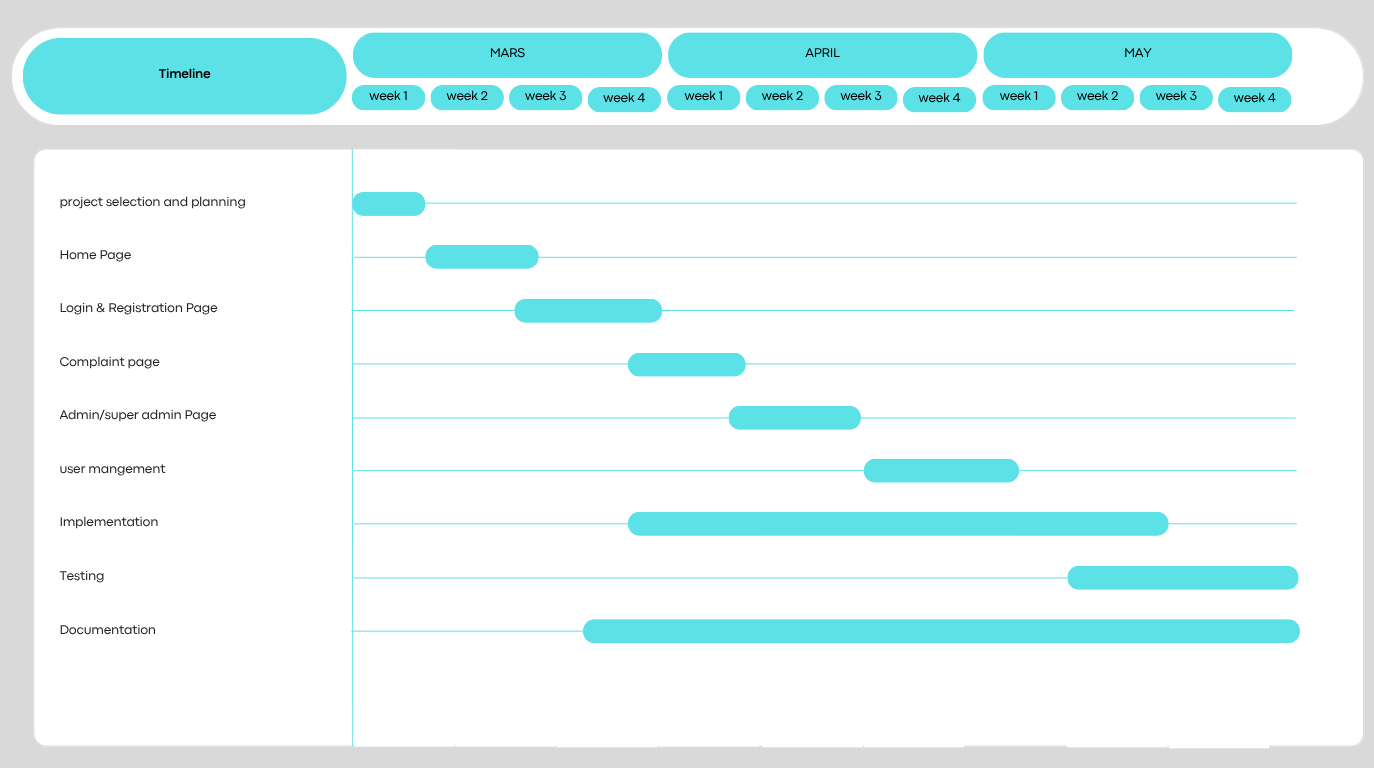


Figure Gantt Chart

# 1.7 Project outline

* **Chapter 2:** compares the system with other existing systems, highlighting the unique functionalities and important features in TheWBCwebsite.
* **Chapter 3:** discusses the project's approach, functional and non-functional needs, and the feasibility assessment.
* **Chapter 4**: explains the system's functionality using a series of graphs for clarity.
* **Chapter 5**: examines the implementation and evaluation of the outcomes.
* **Chapter 6**: summarizes the project idea and the future work that can be applied in the near future.

**CHAPTER 2**

**LITERATURE REVIEW**

# 2.1 Overview

In this chapter, previous studies related to complaint management systems will be reviewed through resources such as Google Scholar, academic journals, websites, and other reputable sources. This review aims to understand existing research on electronic complaint management, common challenges, and solutions applied in universities and similar institutions.

# 2.3 Related Work

The use of digital systems for complaint management has gained significant attention in recent years, especially in educational institutions. This section reviews various complaint management systems, focusing on their design approaches, features, and challenges such as user accessibility, data privacy, and response efficiency. Understanding these existing solutions helps in developing a reliable and effective complaint management system tailored to the needs of the university.

## 2.3.1 Discussions

Many studies and projects have focused on improving complaint management in organizations, especially universities. Previous systems mostly aimed to digitize the complaint submission and tracking process to make it faster and more transparent. Some solutions included online portals where users can submit complaints, while others integrated automated notifications to keep users updated on the status of their complaints.

However, many existing systems face challenges like ensuring data privacy, providing easy access for all users, and handling high volumes of complaints without delays. Also, communication between the complainant and the administrators is often limited to status updates only, such as marking a complaint as "resolved" or "reject" In many cases, users cannot easily request to reopen a complaint or add comments after closure, which reduces interaction and user satisfaction.

Therefore, there is a need for a more interactive system that allows users to communicate through comments, request reopening of complaints if they feel the issue is not fully resolved, or provide feedback when a complaint is closed. These discussions show that while progress has been made, there is still room for improvement in developing a more user-friendly, secure, and efficient complaint management system tailored to the specific needs of universities.

## 2.3.2 Issues and challenges faced by the other project

Many previous complaint management systems have faced several issues and challenges. One common problem is ensuring data privacy and security, as complaints often contain sensitive information that must be protected from unauthorized access.

Another challenge is providing easy and fair access for all users, including students, faculty, and staff, regardless of their technical skills or device availability. Some systems also struggle to handle a large number of complaints efficiently, which can lead to delays in processing and resolving issues.

Communication is another area of concern. Many systems offer limited interaction between users and administrators, often restricting feedback to basic status updates without allowing follow-up comments or reopening requests. This can cause frustration for users who feel their complaints have not been fully addressed.

Additionally, technical issues like server downtime, system crashes, or unreliable internet connections can disrupt service during critical times, reducing the effectiveness of the complaint management process.

These challenges highlight the need for a robust, secure, and user-friendly complaint management system that can handle large volumes, ensure privacy, and facilitate effective communication.

Table System Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature/Aspect |  |  |  |  |
| Allows supervisor tracking and support | Yes | No | No | No |
| Email notifications | Yes | Yes | Yes | No |
| Categorizes complaints into different types | Yes | No | No | No |
| Provides reports and analytics for complaints | Yes | No | No | No |
| Ensures data security and privacy | Yes | Yes | Yes | Yes |

# 2.4 Summary

This section summarized the main features of the complaint management system. The system allows supervisors to track and support the complaint handling process effectively. It sends email notifications to keep users updated on the status of their complaints. Complaints are categorized into different types to improve organization and management.

Additionally, the system provides detailed reports and analytics to help monitor trends and performance. Data security and privacy are ensured to protect sensitive information throughout the process.

These features aim to create a reliable, efficient, and user-friendly system tailored to the needs of the university community.

**CHAPTER 3**

**METHODOLOGY**

# 3.1 Overview

In this chapter, we will discuss the feasibility study related to the complaint management system for the university. This study analyzes the technical, operational, and economic feasibility of implementing the system. Additionally, potential risks that could affect the project’s success will be evaluated using a risk matrix to identify and assess possible challenges.

Moreover, this chapter will outline the system requirements, dividing them into functional requirements (what the system should do) and non-functional requirements (how the system should perform).

# 3.2 Feasibility Study

The real cost for our project is as detailed below in Table 3.1

Table Feasibility Study

|  |  |
| --- | --- |
| Item | Estimated cost |
| |  | | --- | | **Software development tools and licenses** |  |  | | --- | |  | | |  | | --- | | $50 |  |  | | --- | |  | |
| |  | | --- | | **Training and learning materials** |  |  | | --- | |  | | |  | | --- | | $30 |  |  | | --- | |  | |
| Total | **$80** |

The real cost if we applied this project in Jordan below in Table 3.2

Table Real cost for our project

|  |  |
| --- | --- |
| Item | Estimated cost |
| |  | | --- | | **Servers, data storage, and backup** | | |  |  | | --- | --- | | |  | | --- | | $2,000 | | |
| |  |  |  | | --- | --- | --- | | |  | | --- | | **Software licenses and integration tools** |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | $1,000 |  |  | | --- | |  | |  |  | | --- | |  | |
| |  | | --- | | **Security testing, encryption, and system audits** |  |  | | --- | |  | | |  | | --- | | $1,500 |  |  | | --- | |  | |
| |  | | --- | | **System maintenance, troubleshooting, and support** |  |  | | --- | |  | | |  | | --- | | $800 |  |  | | --- | |  | |
| |  | | --- | | **Training university staff and administrators** |  |  | | --- | |  | | |  | | --- | | $300 |  |  | | --- | |  | |
| |  | | --- | | **Total** |  |  | | --- | |  | | **$5,600** |

# 3.3 Methodology

We used the Agile methodology to develop our complaint management system. Agile allows us to work in small, manageable parts called “sprints,” usually lasting two to four weeks. Each sprint focuses on building specific features of the system.

Our team worked closely together and with stakeholders to gather feedback regularly. This helped us make continuous improvements and quickly fix any problems. Agile also promotes flexibility, so we could adapt to changes or new requirements during the project.

The process started with planning the features for the first sprint, then designing, coding, and testing those features. After each sprint, we reviewed what was done, gathered feedback, and planned the next sprint accordingly.

Using Agile helped us deliver a working system step-by-step while keeping the project aligned with the university’s needs.

## 3.3.1 Requirements

**The type(s) of collected the requirements (Interview)**

Interviews were conducted with students, faculty, and administrative staff as part of the requirements gathering process to develop the university complaint management system. During these interviews, the team aimed to understand users’ needs and expectations from the system, focusing on important features and how the system could effectively manage complaints. Guided questions explored how users currently handle complaints and the challenges they face in submitting and resolving them. They also shared their opinions on the user interface and personal preferences for system design. These interviews played a crucial role in guiding development to better meet user expectations and ensure a satisfactory and efficient user experience. (Rezzky, 2021)

**Types of Requirements**

**Functional**

These describe the specific services and functions the complaint management system should provide, including how it should respond to user actions and inputs. Functional requirements may also specify what the system should not do to ensure proper operation.

Table Functional

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| User Login | The system must allow users to log in using their credentials. |
| Profile Management | Users must be able to view, and edit their profiles, including skills and project preferences. |
| Team Creation | Users must be able to create teams, specifying project type, skills required, and programming languages. |
| Team Browsing | Users must be able to browse existing teams based on filters such as project type, and languages. |
| Join Requests | The system must allow students to send a request to join teams. |
| Add Requests | The system must allow team leader to send a request to add students to his team. |
| Admin Panel | Administrator must be able to monitor and manage teams, delete teams, accept/reject teams creation requests and members exit/expulsion requests. |
| Notifications | The system must notify users of important actions, such as join requests and requests to refuse to establish a team. |
| Search for a student | The system must allow students to search for a student using students name or ID. |
| Search for a team | The system must allow students to search for a team using team name. |
| Edit team info | The system must allow team leader to edit team info. |
| Edit deadline | The system must allow administrator to edit the deadline. |

**Non-Functional**

There are constraints on the services or functions offered by the university complaint management system. These include timing constraints that the system must meet for quick response, constraints on the development process to ensure compatibility with the university environment, and constraints imposed by standards related to data protection and privacy. Non-functional requirements usually apply to the system as a whole rather than individual features or services.

Table Non-functional Requirements

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| Usability | The system must have a user-friendly interface, ensuring ease of navigation and interaction. |
| Performance | The platform must handle up to a specified number of concurrent users without performance degradation. |
| Scalability | The system must be designed to accommodate additional users and features in the future. |
| Security | User data must be securely stored. |
| Availability | The platform must maintain a high level of uptime, ensuring reliability for users. |
| Compatibility | The application must be accessible across modern web browsers and devices. |
| Maintainability | The system must be developed using modular code to facilitate future updates and bug fixes. |

## 

**Tools**

* Office365.
* Visual Studio 2022.
* Draw io , ERPlus.
* **Language**: HTML, CSS, JS, Bootstrap, Asp.Net Core, Sql Server Management Studio.
* **AI**: ChatGPT, GitHub Copilot.
* **Web-Browser**: Google chrome / Brave.
* **Hardware**: laptop

## 3.4 Methodology Process

Scrum is one of the most popular software development methodologies. It deals with many environmental and technical variables, such as requirements and resources, which can change during the development process. This can make the process unpredictable and complex if not managed properly. Therefore, the **WCH** system requires flexibility in its development process to quickly and effectively adapt to these changes.

Additionally, Scrum helps improve engineering practices within the team by involving frequent management activities aimed at identifying any problems or obstacles in the development process and the practices used. The **WCH** system is developed using Scrum because it is one of the most widely used and trusted frameworks in the software industry.

It offers easy scalability, flexibility to changes, improved software quality, and helps reduce risks during development.

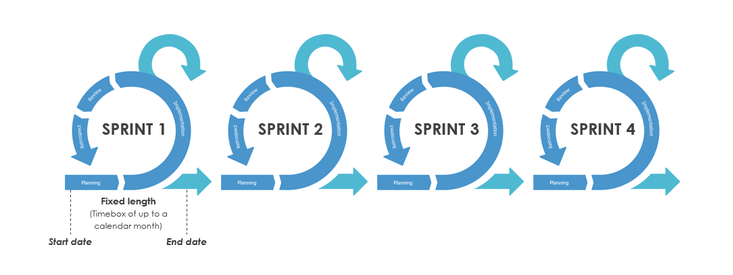


Figure Sprints Process

**Main Roles in Scrum**

* **Scrum Master**  
  Responsible for facilitating the Scrum process and ensuring the team follows the methodology correctly.
* **Product Owner**  
  Responsible for defining project requirements and prioritizing the work.
* **Scrum Team**  
  The development team that executes the work and collaborates to achieve project goals.
* **Customer**  
  The user or stakeholder who benefits from the final product and provides feedback.
* **Management**  
  The management team that supports the Scrum team by providing resources and guidance.

**Sprints**

A sprint is an iterative development cycle during which system functionalities are developed or improved to produce a new updated increment of the system. Each sprint goes through the traditional software development phases: requirements gathering, analysis, design, development, and delivery. During the sprint, the system’s architecture and design gradually evolve as new features are added or existing ones are enhanced.

A sprint typically lasts from one week to one month and is planned in advance. In some cases, more than one team may work simultaneously on building the increment.

Table Scrum Sprints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Duration** | **Objectives** | **Tasks** | **Deliverables** |
| **Sprint 1** | 2 Weeks | 1. Identify and define the problems students face when forming and managing teams.  2. Understand user needs for the complaint management system.  3. Create initial system diagrams. | 1. Conduct interviews with students to gather insights.  2. Review existing complaint systems and best practices from other universities.  3. Create Use Case Diagrams and Data Flow Diagrams (DFDs). | - Problem Statement & System Requirements  - Functional & Non-Functional Requirements List |
| **Sprint 2** | 5 Weeks | 1. Design the system architecture and structure.  2. Select suitable development technologies. | 1. Design the Entity-Relationship Diagram (ERD).  2. Create UI/UX mockups (registration, login, complaint submission).  3. Choose the technology stack:  ▸ Frontend: Bootstrap, CSS, JavaScript  ▸ Backend: ASP.NET Core  ▸ Database: SQL Server Management Studio 20  4. Set up the development environment and localhost testing. | - Finalized System Design (ERD, UI/UX)  - Configured Development Environment |
| **Sprint 3** | 6 Weeks | 1. Build core platform features.  2. Conduct iterative testing and refinements. | 1. Implement user authentication (login, password management).  2. Develop features: complaint submission, status tracking, comments, and reopen requests.  3. Integrate database with user and complaint data.  4. Perform unit testing.  5. Conduct team code reviews and gather user feedback. | - Working Prototype with Core Features  - Iterative Testing Results and Improvements |
| **Sprint 4** | 2 Weeks | 1. Validate the platform’s functionality and usability.  2. Prepare for deployment and future scalability. | 1. Conduct system, usability, and functional testing on localhost.  2. Perform user testing with students.  3. Collect feedback from Dr. Noor Al-Qudah and student users.  4. Document deployment process for live server.  5. Finalize full project documentation: Acknowledgements, Abstract, Problem Statement, Objectives, and Research Strategy. | - Fully Functional & Validated Platform  - Deployment Documentation  - Final Project Report |

# CHAPTER 4

# DESIGN MODELS

# 4.1 Overview

This chapter presents the design models for the WISE Complaint Hub system, a web-based complaint management platform designed for The World Islamic Sciences and Education University. The system aims to provide a secure, transparent, and efficient platform for students and university staff to submit, manage, and resolve institutional complaints. The design models in this chapter include the **Context Diagram, Data Flow Diagram (DFD), Use Case Diagram, Entity-Relationship (ER) Diagram,** and **Relational Model.** These models are essential for understanding the system’s architecture, the flow of information between users and system components, user interactions with various system functionalities, and the underlying database structure that supports data storage and retrieval. Each model provides a unique perspective on the system, ensuring a comprehensive understanding of its design and functionality.

# 4.2 Context diagram-0

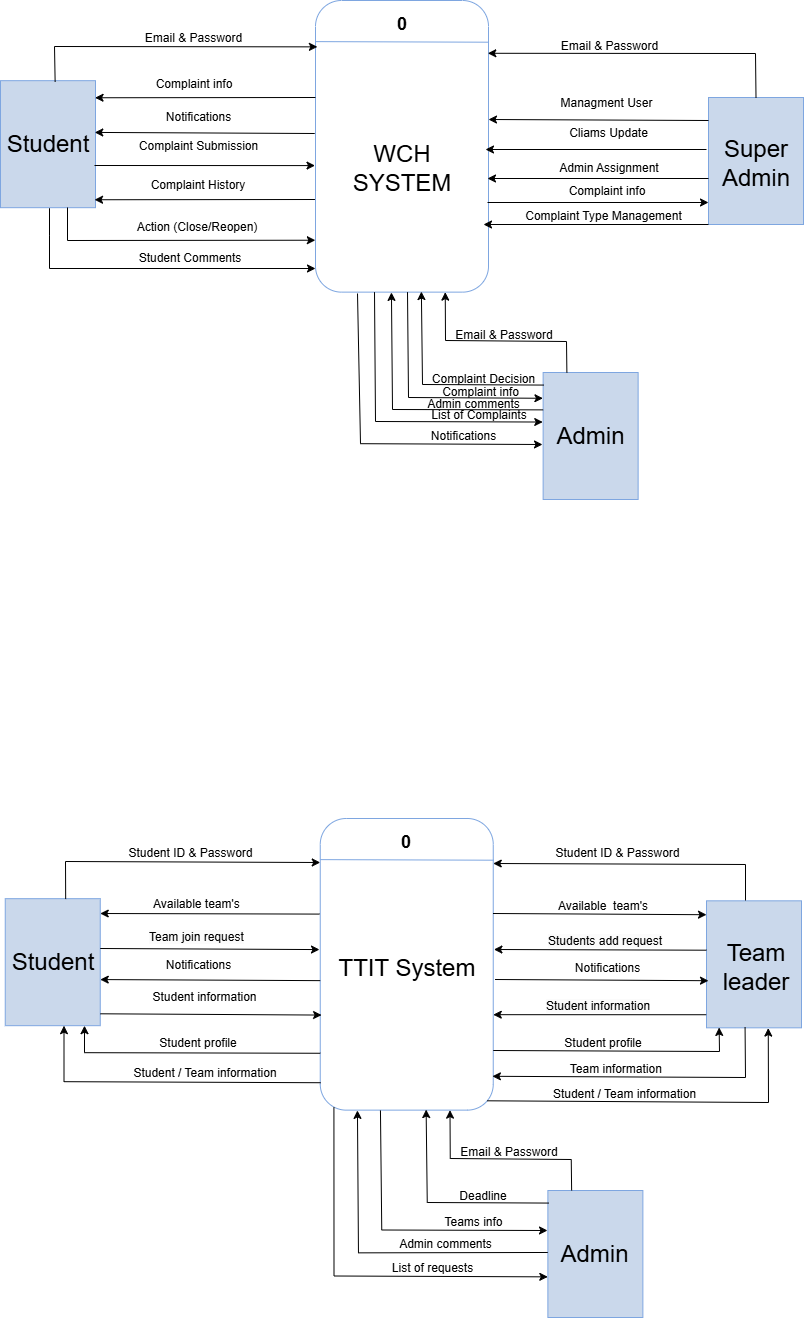
Provides a high-level view of the WISE Complaint Hub system, illustrating the interaction between external entities (students, university staff, and administrators) and the system in Figure 4.1. This diagram outlines the main system boundaries, highlighting how users interact with the complaint management platform to submit complaints, track their status, and receive feedback from the university's resolution team.

Figure Context diagram-

# 4.3 Data flow Diagram-1

Depicts the main processes within the **WISE Complaint Hub** system and the flow of data between these processes and external entities in **Figure 4.2.** The diagram illustrates how complaints are submitted by students, processed by the system, and managed by university administrators, ensuring a seamless flow of information for complaint resolution.

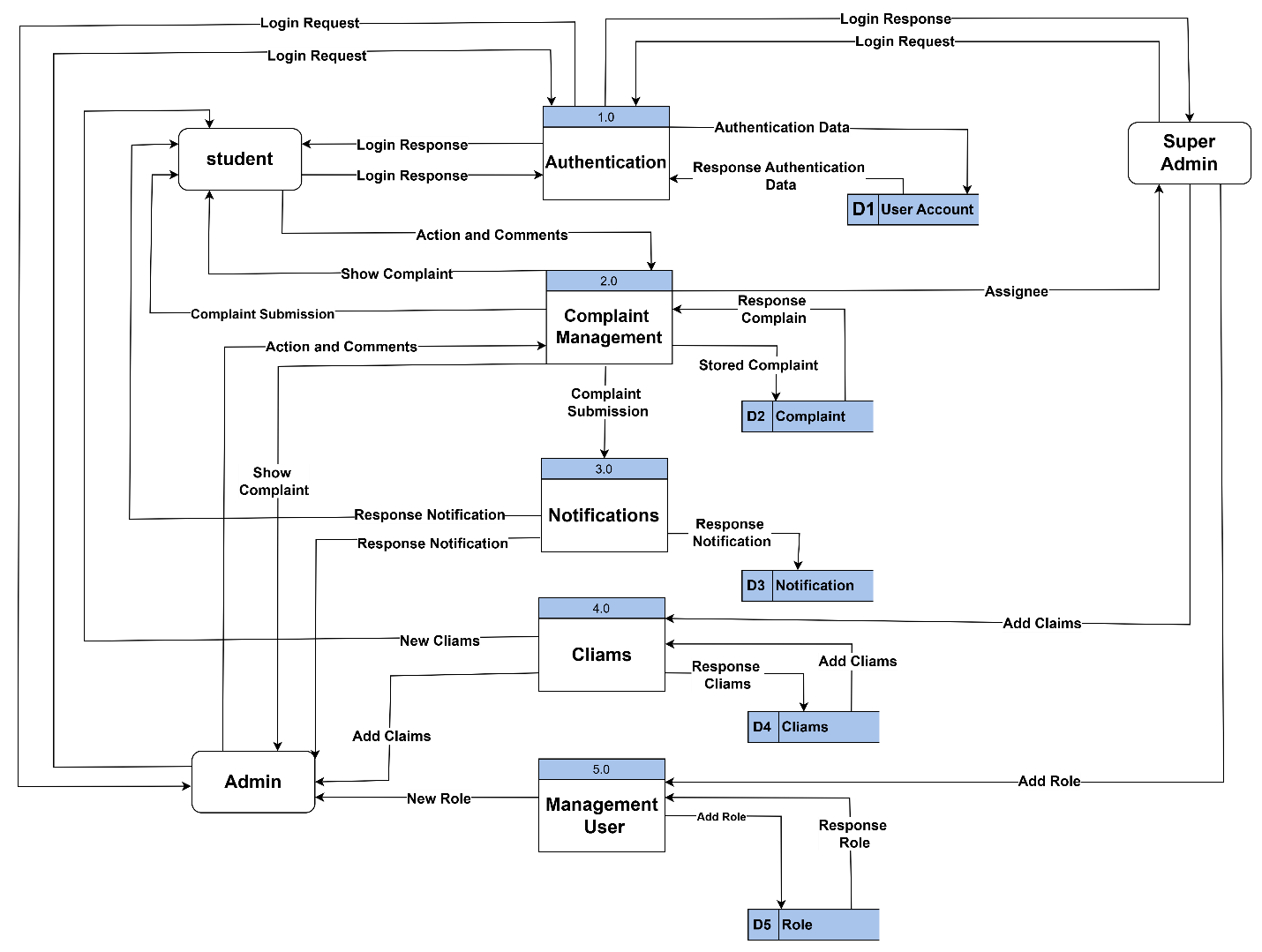


Figure Data flow Diagram

# 4.4 Use Case Diagram

The **Use Case Diagram** for the **WISE Complaint Hub** system is shown in **Figure 4.3**. It illustrates the main interactions between the system and users, highlighting key functionalities such as complaint submission, tracking, notifications, and role management.

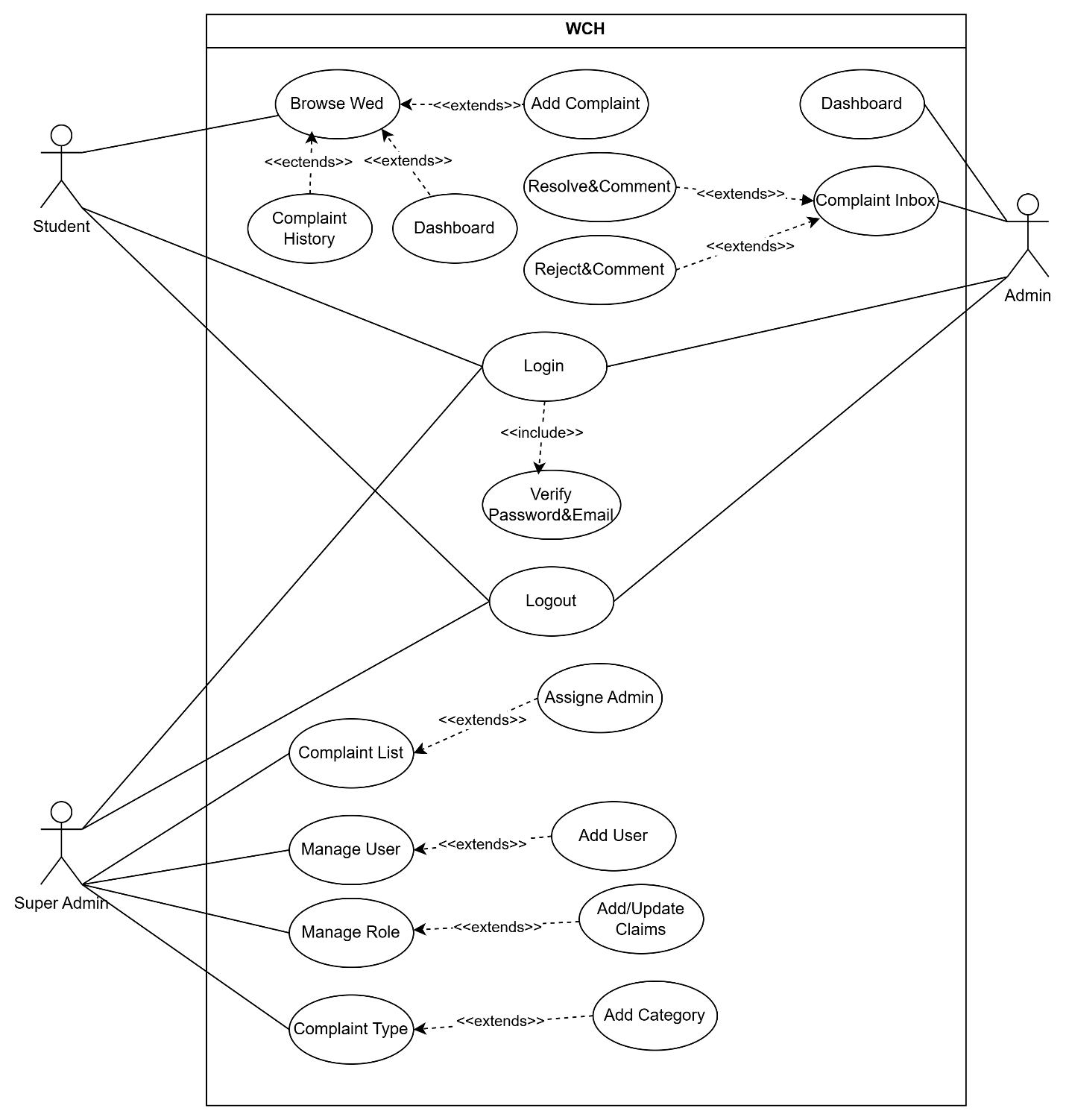


Figure Use Case Diagram

# 4.**5 ER Diagram**

The **ER Diagram** in **Figure 4.4** illustrates the data entities, their attributes, and relationships within the **WISE Complaint Hub** database. It shows how data such as user accounts, complaints, notifications, and roles are structured and connected.

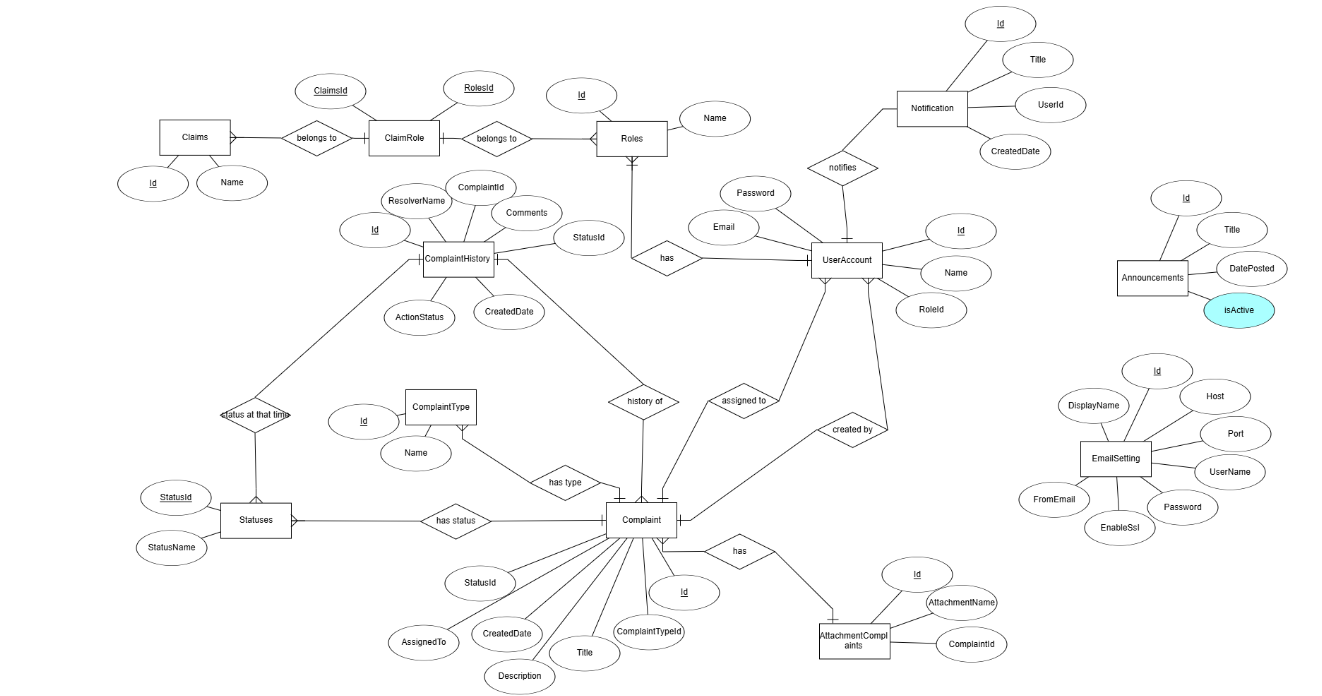
******

Figure 4.4: ER Diagram

# 4.6 Relational Model

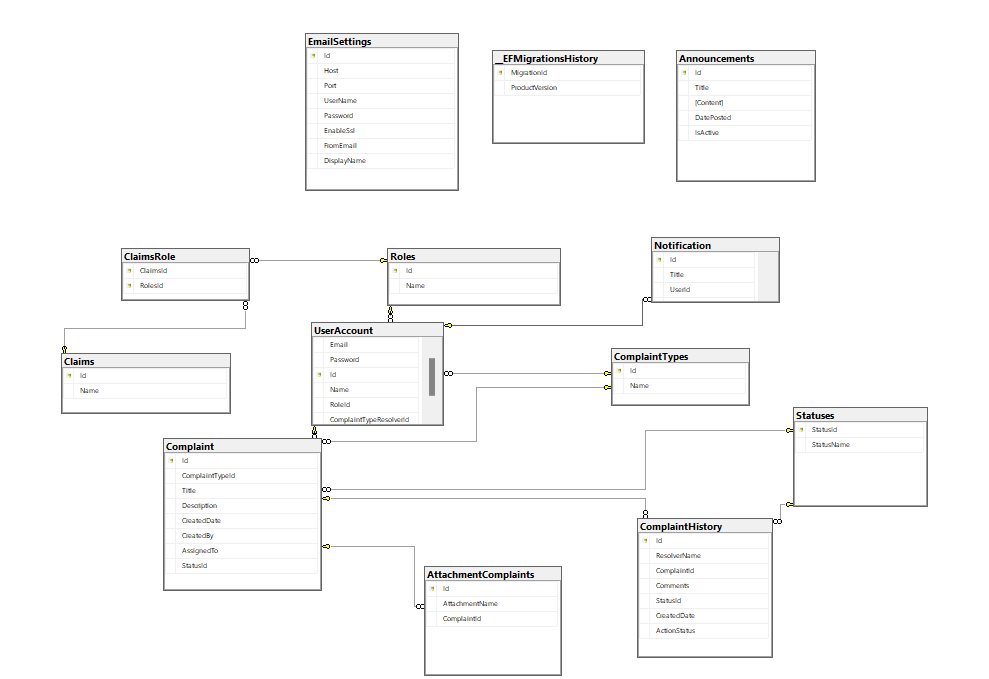
****The Relational Model defines how the entities in the WISE Complaint Hub system are represented as tables in the relational database. It outlines the structure of each table, including fields, primary keys, and foreign key relationships, ensuring logical data organization and integrity. Key tables include Users (with UserID as primary key), Complaints (linked to Users via UserID), Notifications, and Roles. This model supports efficient data management and enforces consistent relationships within the system. Shown in

Figure Relational model

**CHAPTER 5**

**EXPERIMENTS AND RESULTS**

# 5.1 Overview

In this chapter, we will discuss the testing process of the integrated Complaint Management System for University to determine if it meets the specified requirements. We will outline the experiments conducted, the testing methods used, and the results obtained.

The goal is to evaluate the system’s functionality, usability, and reliability in handling student and staff complaints effectively. This includes assessing how well the system supports complaint submission, tracking, assignment, and resolution — as well as how it has proper access control, data integrity, and user satisfaction throughout the complaint lifecycle.

# 5.2 Testing methodologies

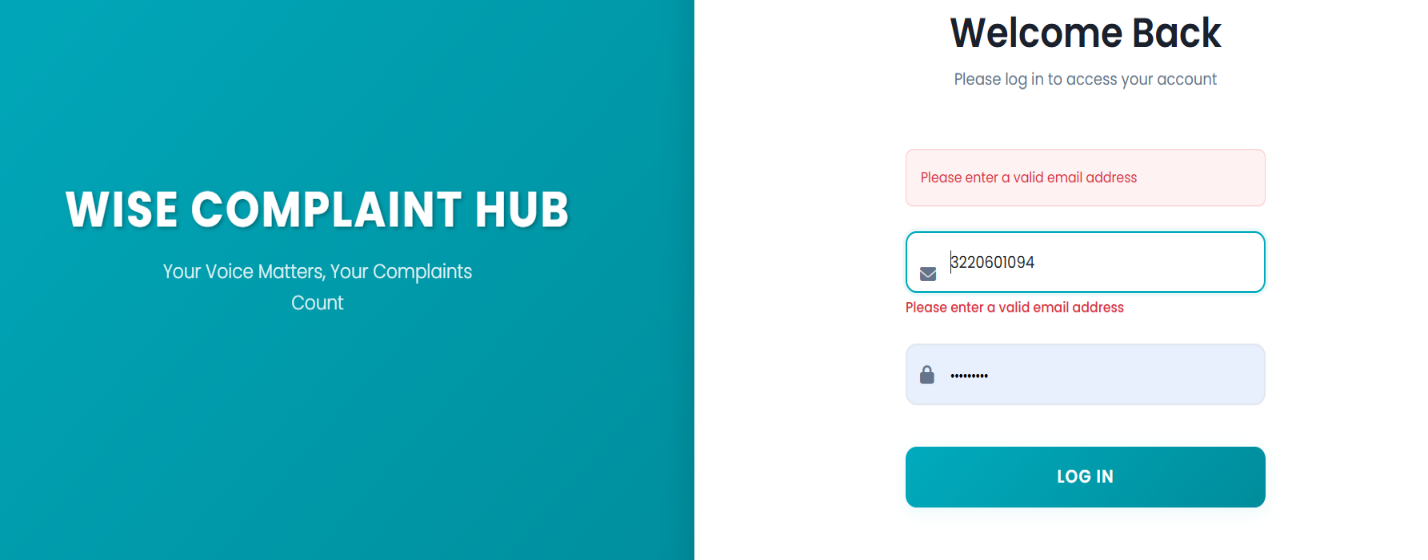
To ensure the reliability and efficiency of the Complaint Management System for University, several testing methodologies were employed throughout the development process. These methodologies helped verify that each module functions as expected and that the system as a whole meets the functional and non-functional requirements.

## 5.2.1 Unit Testing Results

Each component of the system, such as user authentication, complaint submission, and role-based access control, was tested individually to verify correct behavior and error handling.

## 5**.2.1.1 Login:**

Figure Login Page Email & Password

****

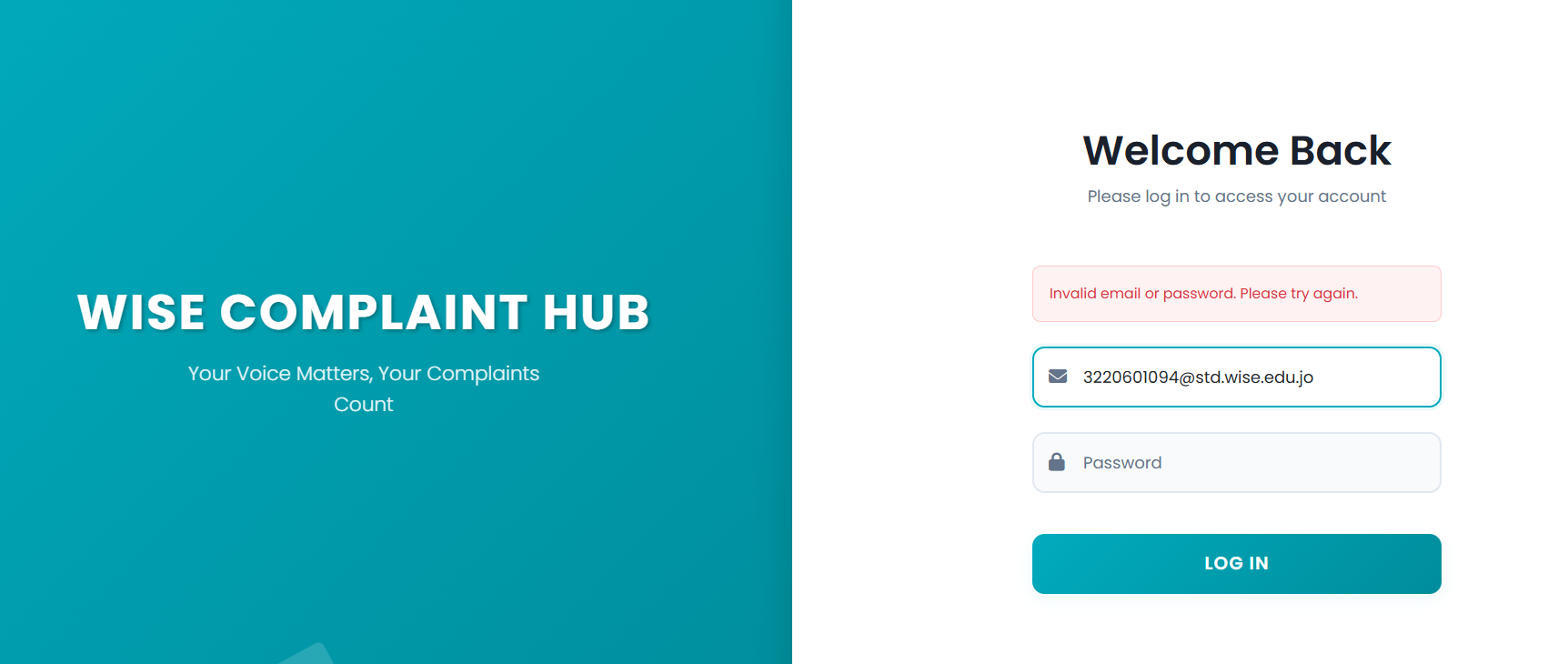


Figure Invalid Email or Password

Figure Validation Please Enter A valid Email Address

Figure must be at least 4 ch password

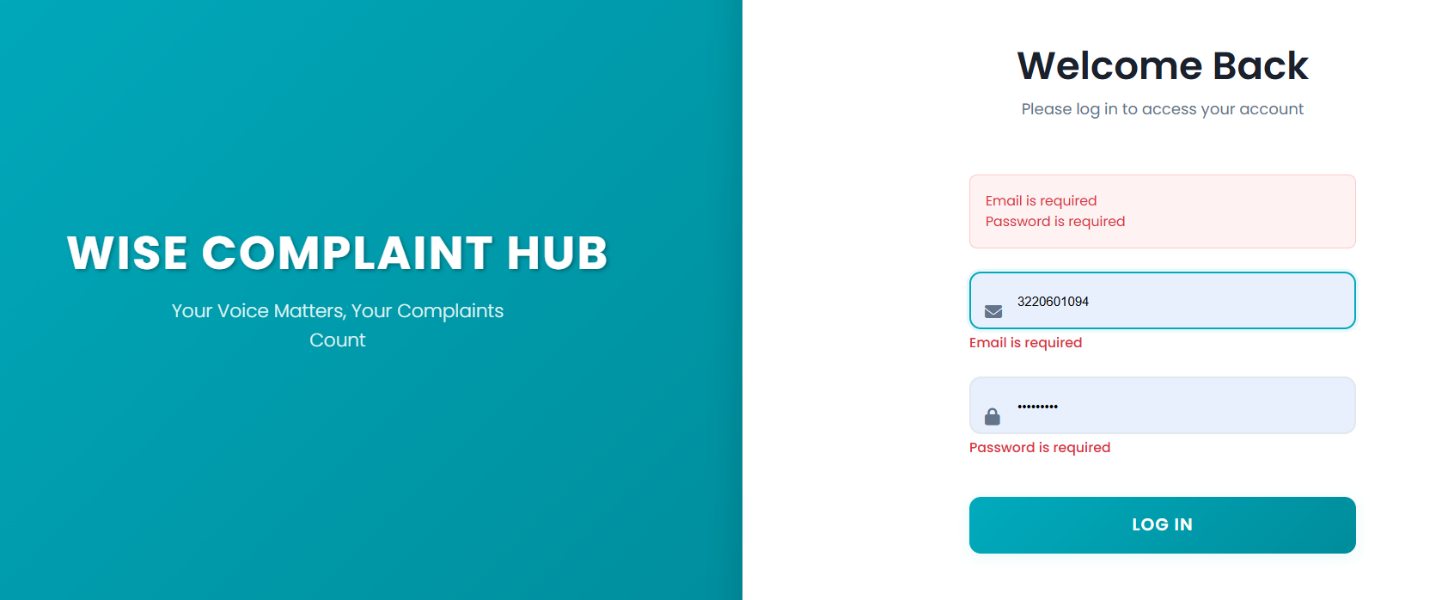
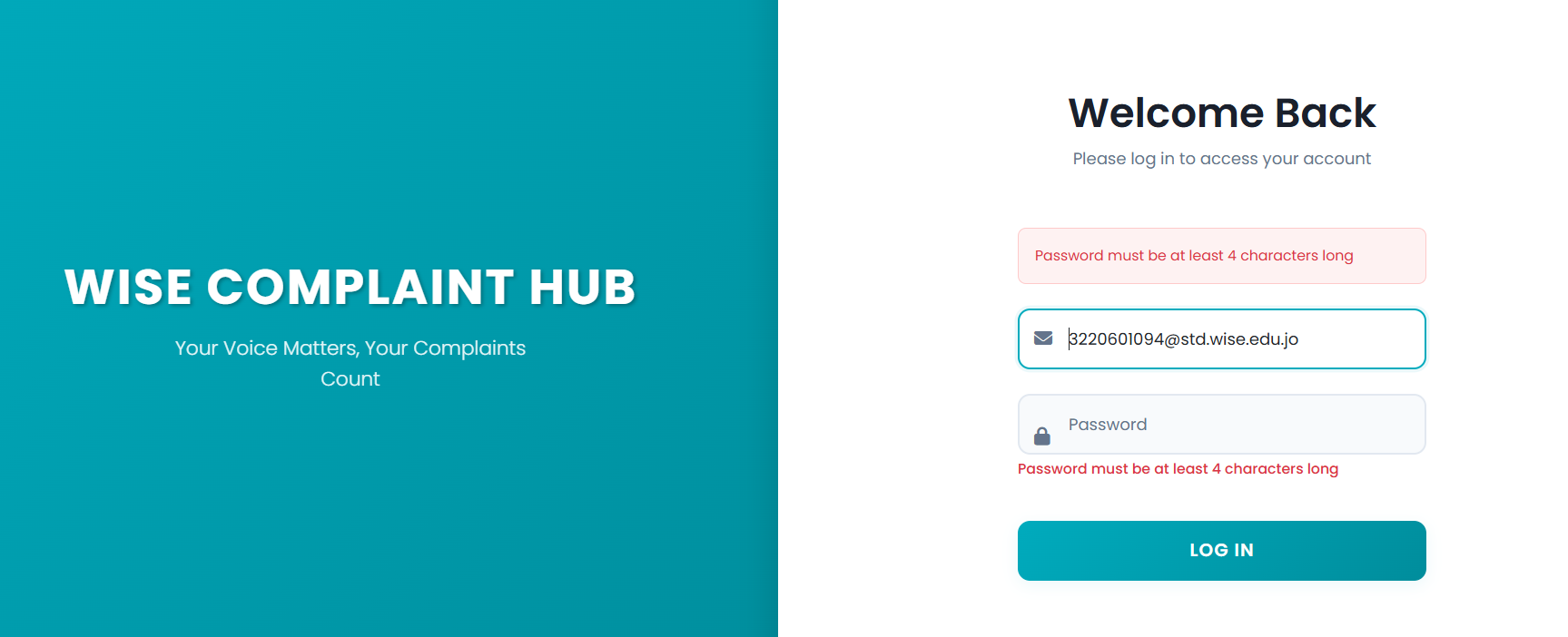


Figure Email and Password Is Required

Figure Complaint submission

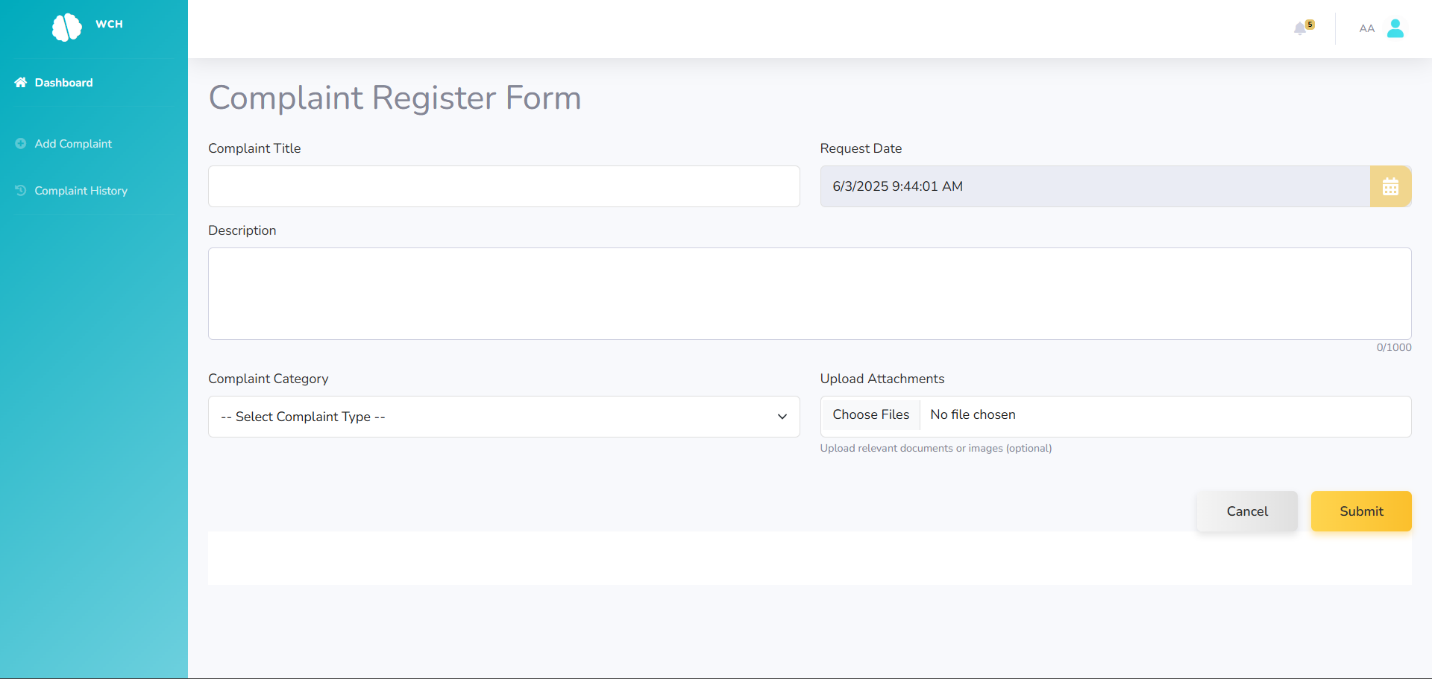
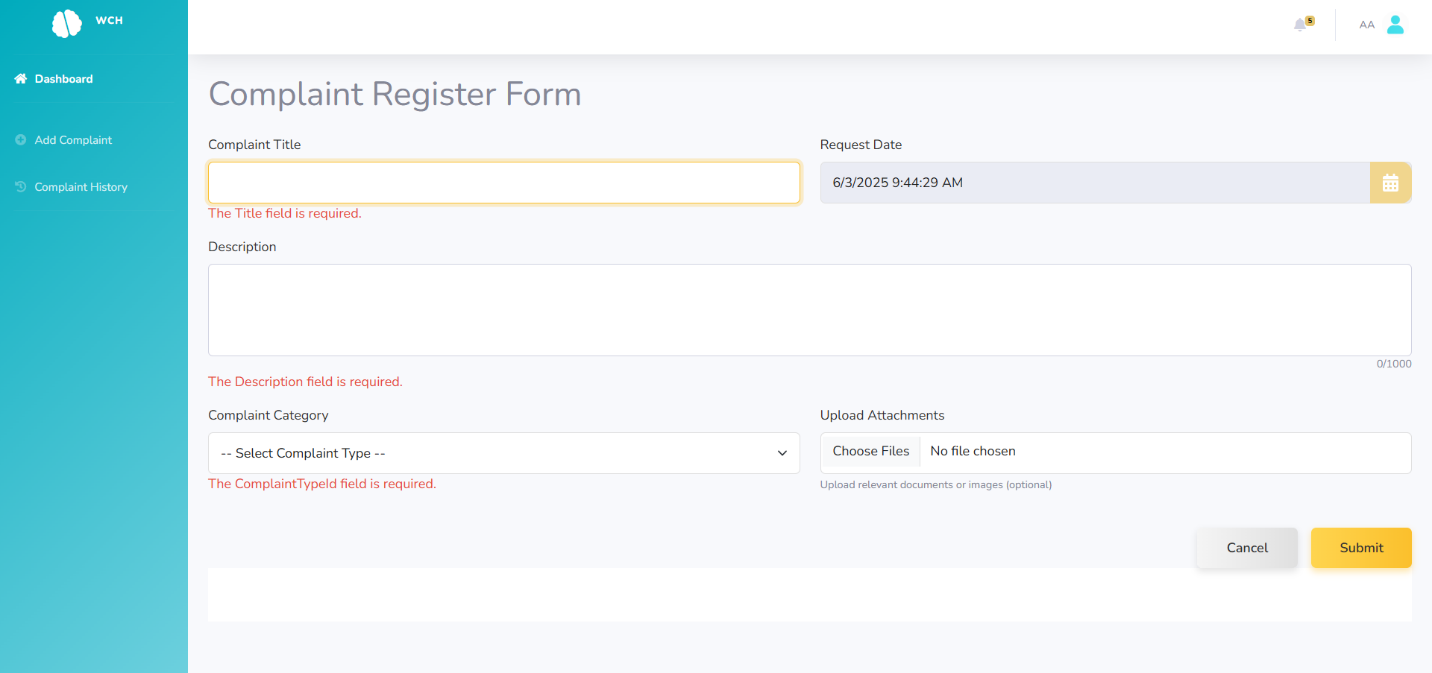
****

Figure Required Title, Description, Catogory

****

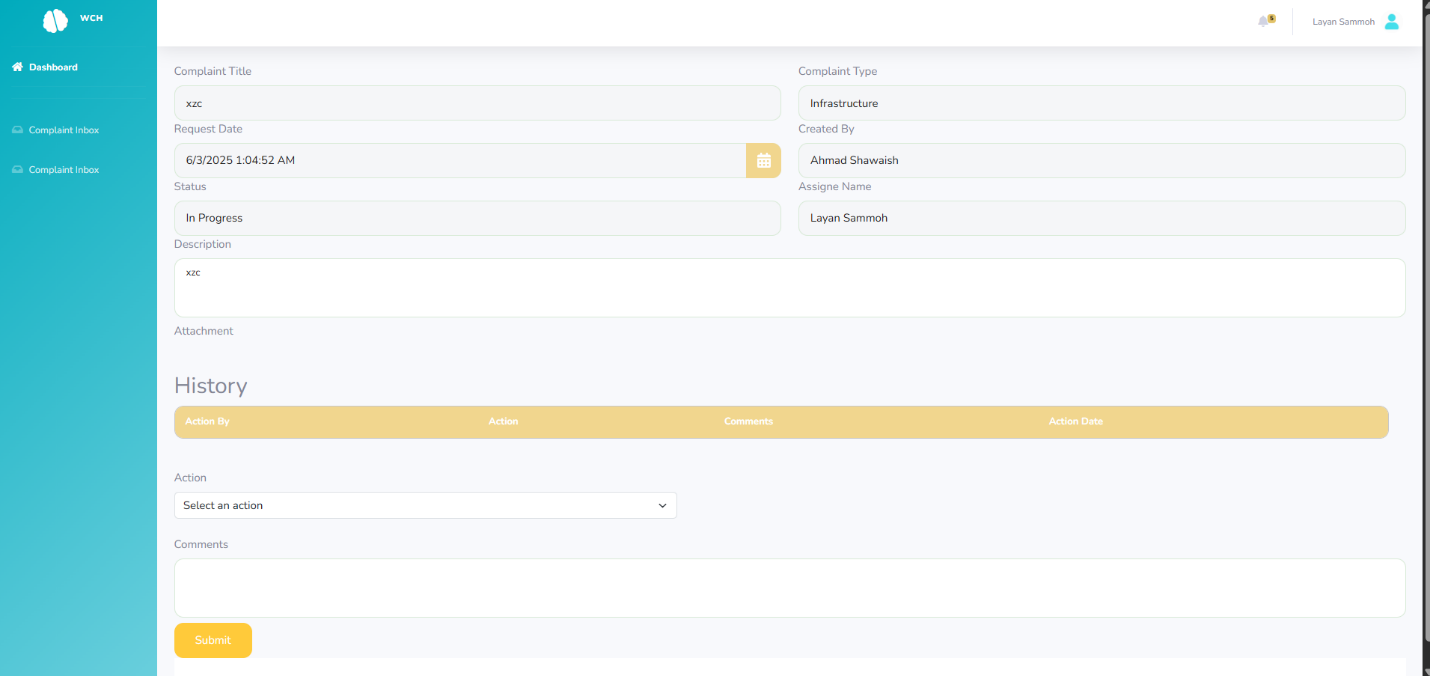
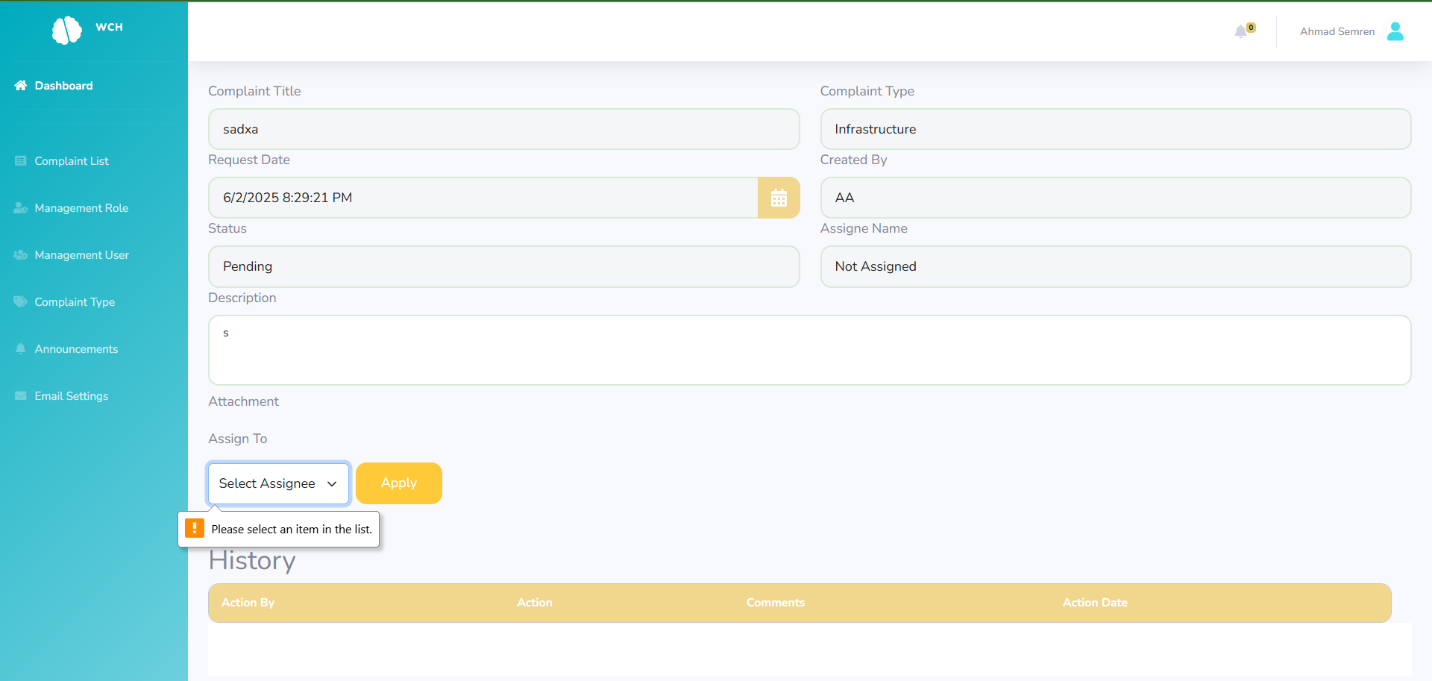
****

Figure Make Assignee (SuperAdmin)

Figure Make Action And Comment (Complaint Handler)

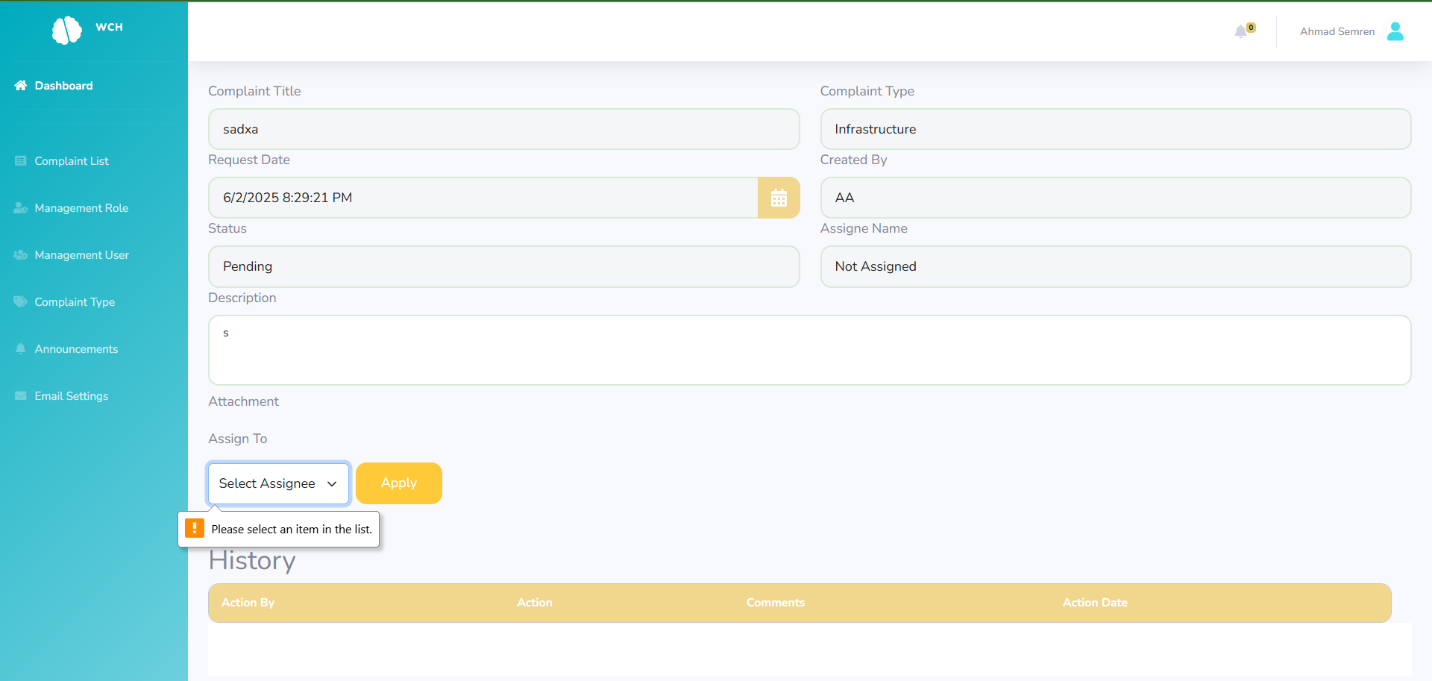
****

Figure select assignee

## 5.2.2 Integration Testing Results

Once individual components were tested, they were integrated and tested together to ensure smooth communication between modules — such as between the complaint submission form and the database, or between user roles and complaint assignments.

## 5.2.3 System Testing Results

System testing was conducted after completing the integration of all modules, including user registration, complaint submission, admin assignment, status tracking, and feedback handling. The purpose was to validate the end-to-end functionality of the system under realistic scenarios.

The testing involved multiple test cases simulating actions from different user roles — students, staff, and administrators. Key areas tested include:

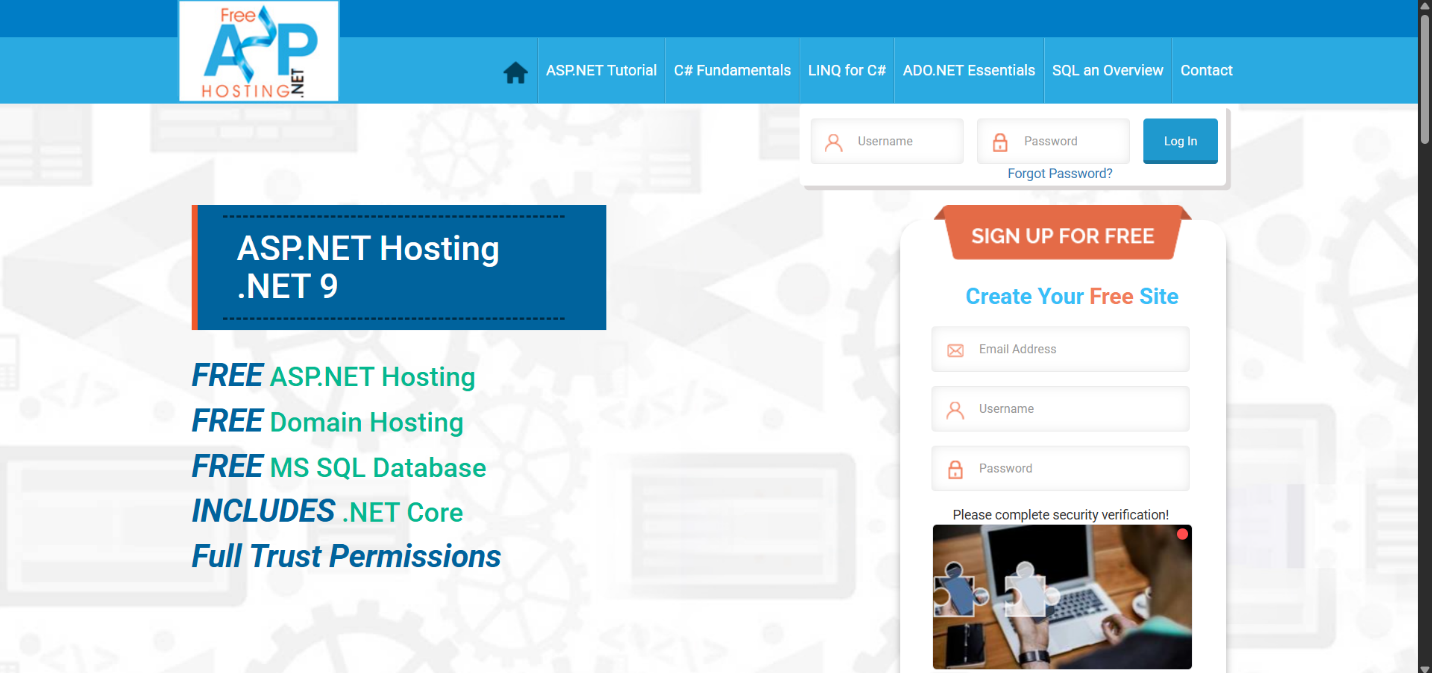
* Successful login/logout for each user role.
* Submission of complaints with attachments.
* Admin viewing and assigning complaints.
* Updating complaint statuses and tracking complaint history.
* Access control based on user roles.

Figure asp.net hosting

## 5.2.4 Acceptance System Results

The system was presented to a group of users (students, faculty, and administrative staff) for acceptance testing. The users were asked to interact with the system by submitting complaints, viewing statuses, and responding to updates. Most users found the system intuitive and functional. Minor suggestions were received for UI improvements. Overall, the system was accepted with positive feedback.

# 5.3 Deployment and Hosting

After completing development and testing, the Complaint Management System was deployed using an ASP.NET hosting service. The application was published from Visual Studio and configured with a live SQL Server database. This allowed users to access the system securely over the internet, test its functionality in a real-world environment, and provide feedback for ******improvements.

# 5.4 Discussion and evaluation

The Complaint Management System for University successfully simplifies complaint submission and tracking for users. Testing showed the system is user-friendly, efficient, and reliable. It handles multiple users and roles without issues, ensuring data accuracy and security through role-based access and validation.

Hosting the project on an ASP.NET server allowed real-time access for testing and use. Overall, the system meets its goals and provides a practical solution for managing university complaints.

# CHAPTER 6

# CONCLUSION AND FUTURE WORKS

# 6.1 Overview

In this chapter, we summarize the project's conclusion, achieved objectives, main contributions, limitations, and potential future work. The Complaint Management System for university students has been evaluated based on its effectiveness in improving transparency, accountability, and efficiency in the handling of student complaints. The system facilitates smooth communication between students and university staff, ensures timely responses, and provides a secure platform for managing and resolving complaints. This chapter also outlines the project’s impact on enhancing student satisfaction and highlights opportunities for future enhancements and scalability.

# 6.2 Summary about the project

In conclusion, the Complaint Management System offers a secure, efficient, and user-friendly platform for managing student complaints at the university. It enhances transparency, accountability, and communication by enabling students to submit, track, and resolve their complaints easily. The system also provides administrators with tools for analyzing complaint data, supporting continuous improvement of university services. Through its design, the system fosters a positive academic environment by streamlining complaint handling and ensuring data privacy.

# 6.3 Achieved objectives

* Achieved providing students with a simple and user-friendly platform to submit complaints easily and securely.
* Achieved enabling real-time tracking of complaint status, ensuring transparency and accountability throughout the resolution process.
* Achieved reducing delays and errors common in manual complaint handling methods by introducing a fully digital system.
* Achieved maintaining organized records of all complaints and generating detailed reports to support data-driven decision-making and continuous service improvement.
* Achieved ensuring data security and privacy for all submitted complaints and user information, adhering to best practices in data protection.

# 6.4 Main contributions of the work

The primary contribution of this project is the development of a secure, efficient, and transparent Complaint Management System tailored for university students. The system significantly improves the complaint handling process by offering a digital platform that simplifies complaint submission, tracking, and resolution. It enhances transparency and accountability through real-time status updates, automated notifications, and detailed reports for administrators. By streamlining processes, the system reduces errors and delays commonly found in manual methods. It also ensures data privacy and protection by incorporating secure authentication and robust access control mechanisms. Overall, the system represents a scalable, user-centric solution that addresses key challenges in managing complaints and fosters a more responsive and student-friendly academic environment.

# 6.5 Limitation

* The system is designed exclusively for the university and does not support complaints related to external institutions or organizations.
* A stable internet connection is required for the system to function effectively.
* The system relies on the university’s servers, which may lead to technical issues or downtime during peak periods.
* The platform is accessible only to users with compatible devices, such as computers or smartphones.
* System administration and complaint management require full access privileges, limiting control to authorized personnel.

# 6.6 Future Work

* Develop a mobile application version to enhance accessibility and user experience, allowing students and staff to manage complaints and appointments conveniently via smartphones.
* Enable students to book appointments directly with relevant departments based on the nature of their complaint (e.g., classroom issues, lecturers), integrating the system with department schedules and staff calendars for efficient real-time booking.
* Implement a user feedback and rating system within the platform to gather suggestions and improve the system continuously based on user experience.
* Integrate advanced analytics and AI technologies for automatic complaint classification and prioritization, improving administrative efficiency and response times.
* Enhance multi-channel notification capabilities (e.g., SMS, email, university portal alerts) to keep users informed throughout the complaint resolution process.
* Expand the system to support complaints from other university affiliates and external institutions, broadening its applicability.
* Add multi-language support to accommodate the diverse university community.
* Improve system scalability and robustness to handle increased user loads and ensure stable performance.
* Implement a user feedback mechanism within the platform to collect suggestions, report issues, and enhance continuous improvement based on user experience.

REFERENCES

**1- Article:**

Author(s), “Article title”, Journal title, vol., no., Page number(s), month-Year.

Ex:

|  |  |
| --- | --- |
| [1] | H. S. A. &. A. E. Al-Shourbaji, "Boosting Ant Colony Optimization with Reptile Search Algorithm for Churn Prediction," *Mathematics,* vol. 10, no. 7, p. 1031, 2022. |

**2- Conference:**

Author(s), “Article title”, Conference title, Country, City, Page number(s), month-Year.

Ex:

|  |  |
| --- | --- |
| [1] | M. R. a. C. A. C. C. Sierra, "Improving PSO-based multi-objective optimization using crowding, mutation and∈-dominance," in *International conference on evolutionary multi-criterion optimization*, Berlin, Heidelberg, 2015. |

**3- Book:**

Author(s), “Book title”, Publisher, Year

Ex:

|  |  |
| --- | --- |
| [1] | S. J. a. H. E. Stein, The EQ edge: Emotional intelligence and your success, Florida: John Wiley & Sons, 2011. |

**4- Website:**

**Author(s), "***Page title."***Website title. Web Address (accessed date retrieved).**

Ex:

[1] D. Holland, "Finding the Building Blocks of Wood."unimelb.edu.au. <https://pursuit.unimelb.edu.au/articles/finding-the-building-blocks-of> wood?utm\_source=linkedin.com&utm\_medium=social&utm\_content=story (accessed June 13, 2021).

**Appendices [if Any]**

Figure dashboard super admin

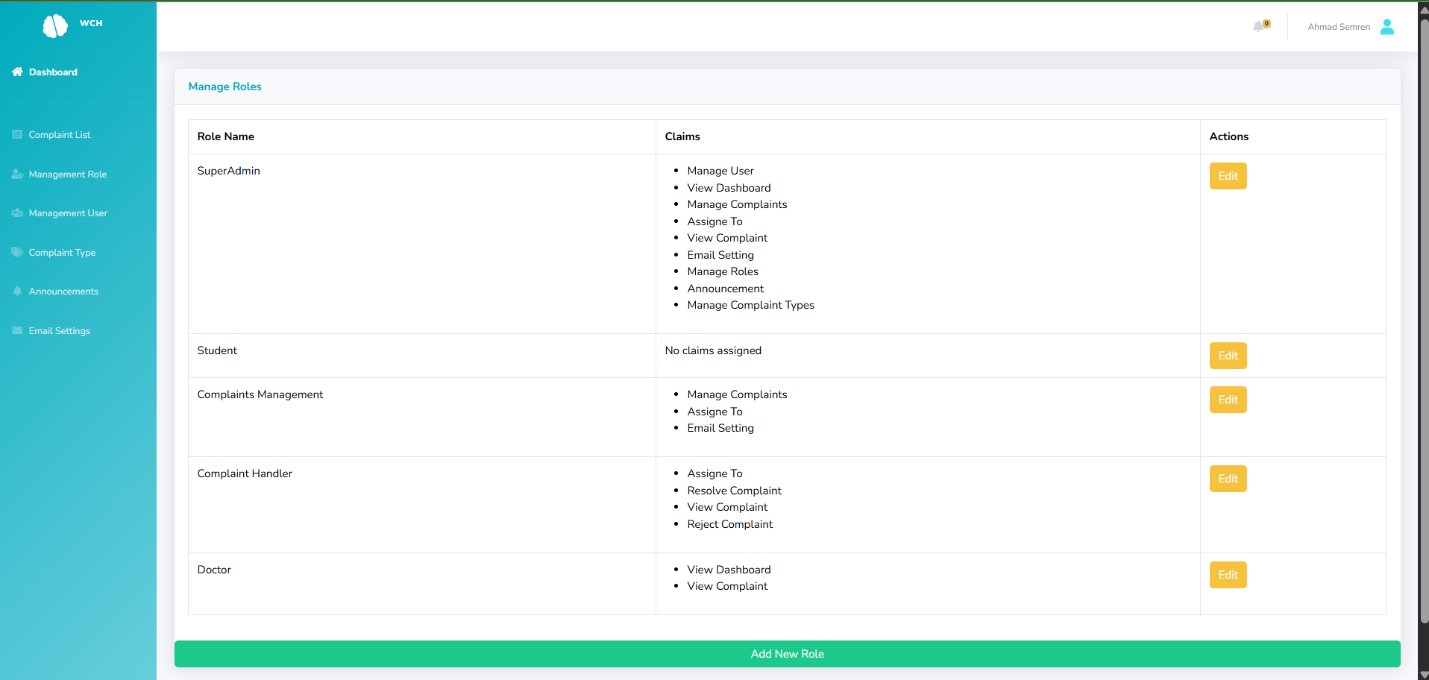
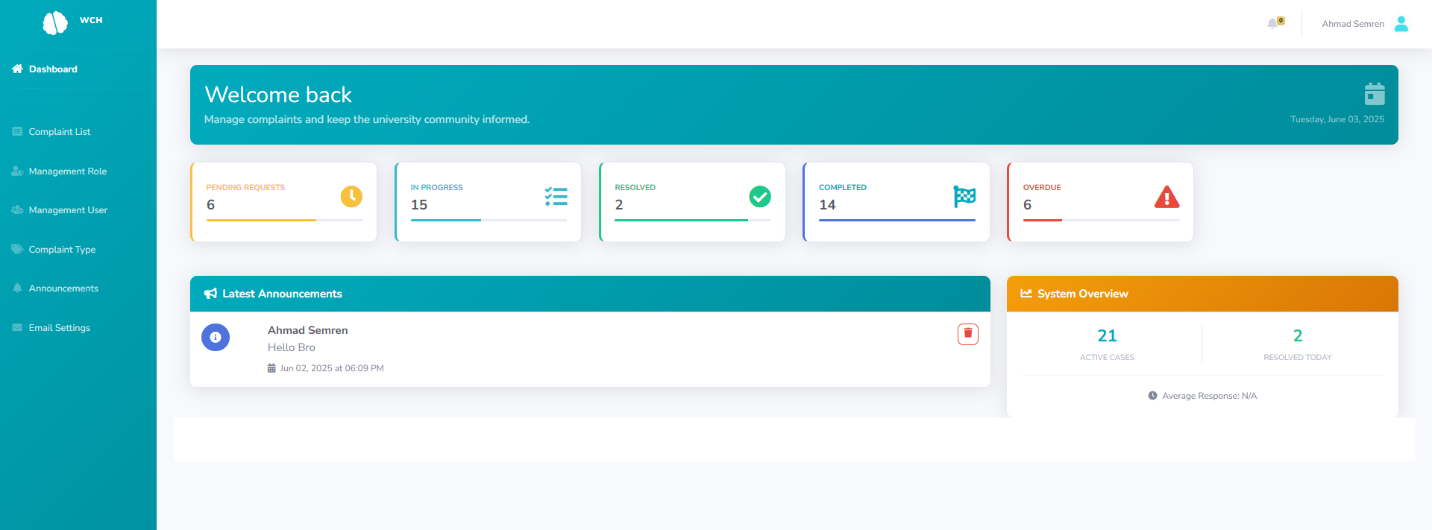


Figure dashboard Student

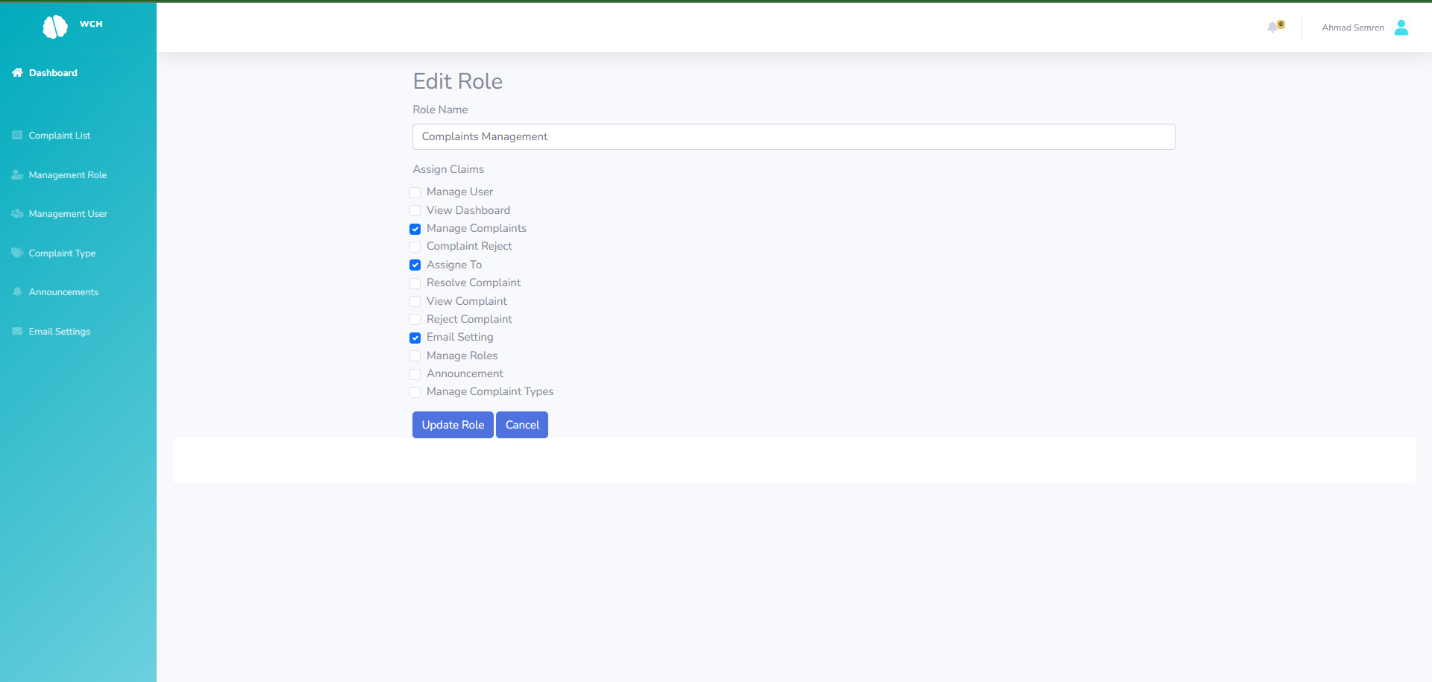


Figure Manges role

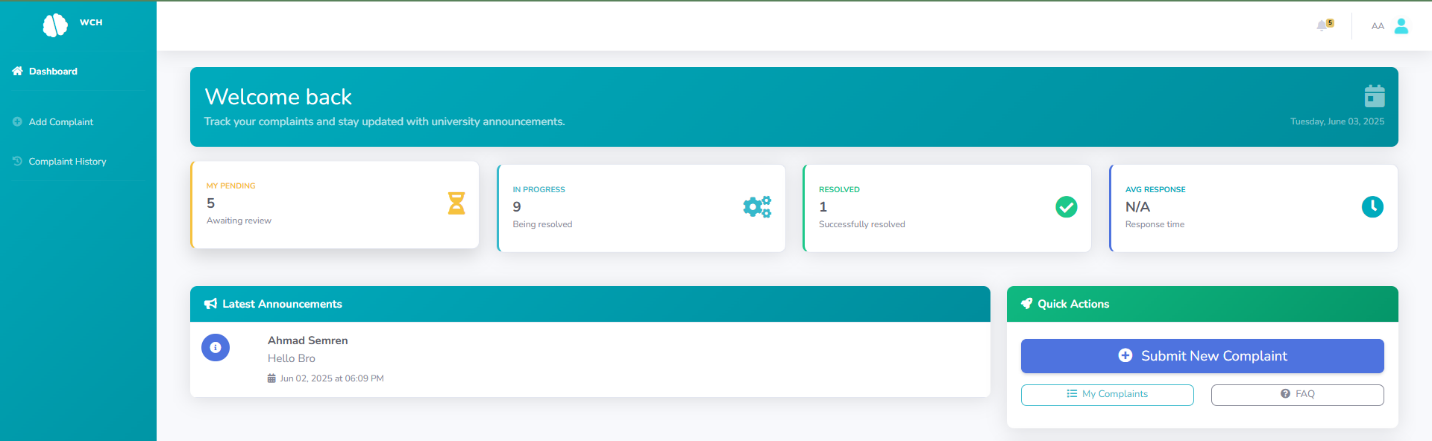


Figure Edit role and Claims

Figure manage user

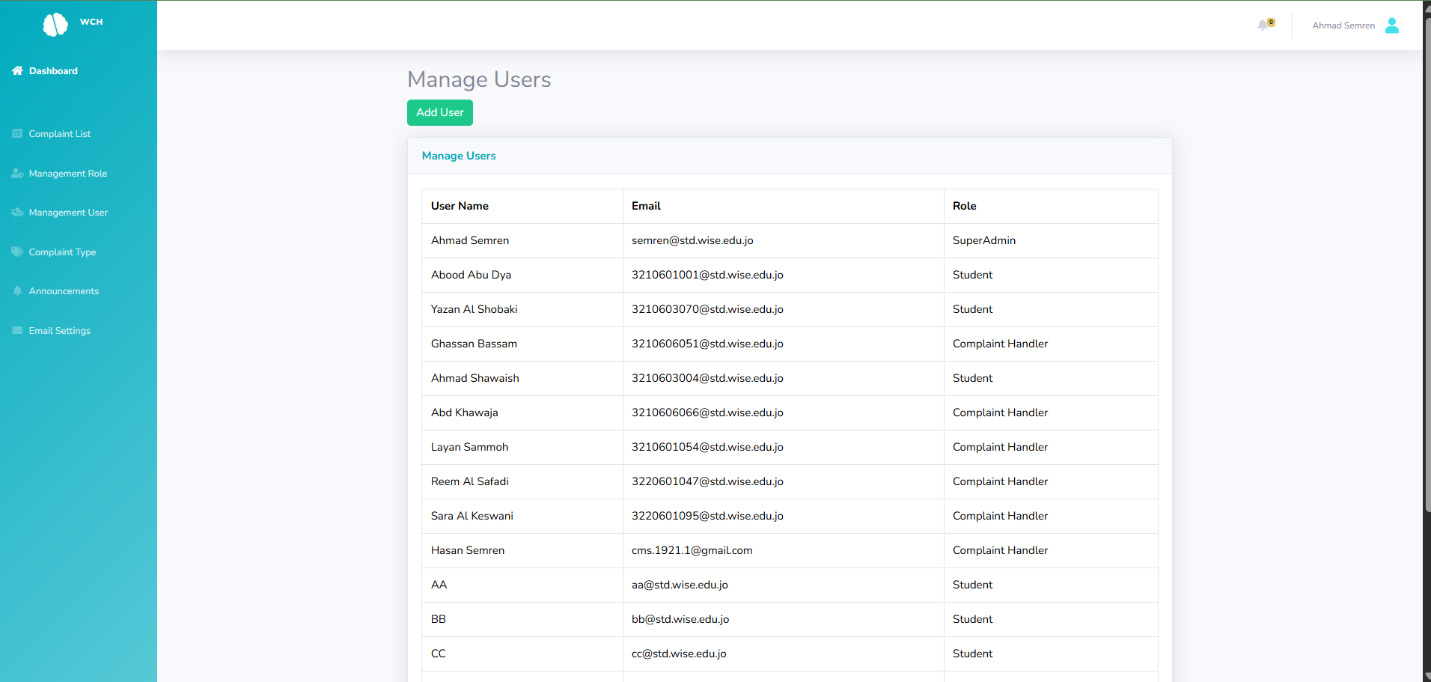
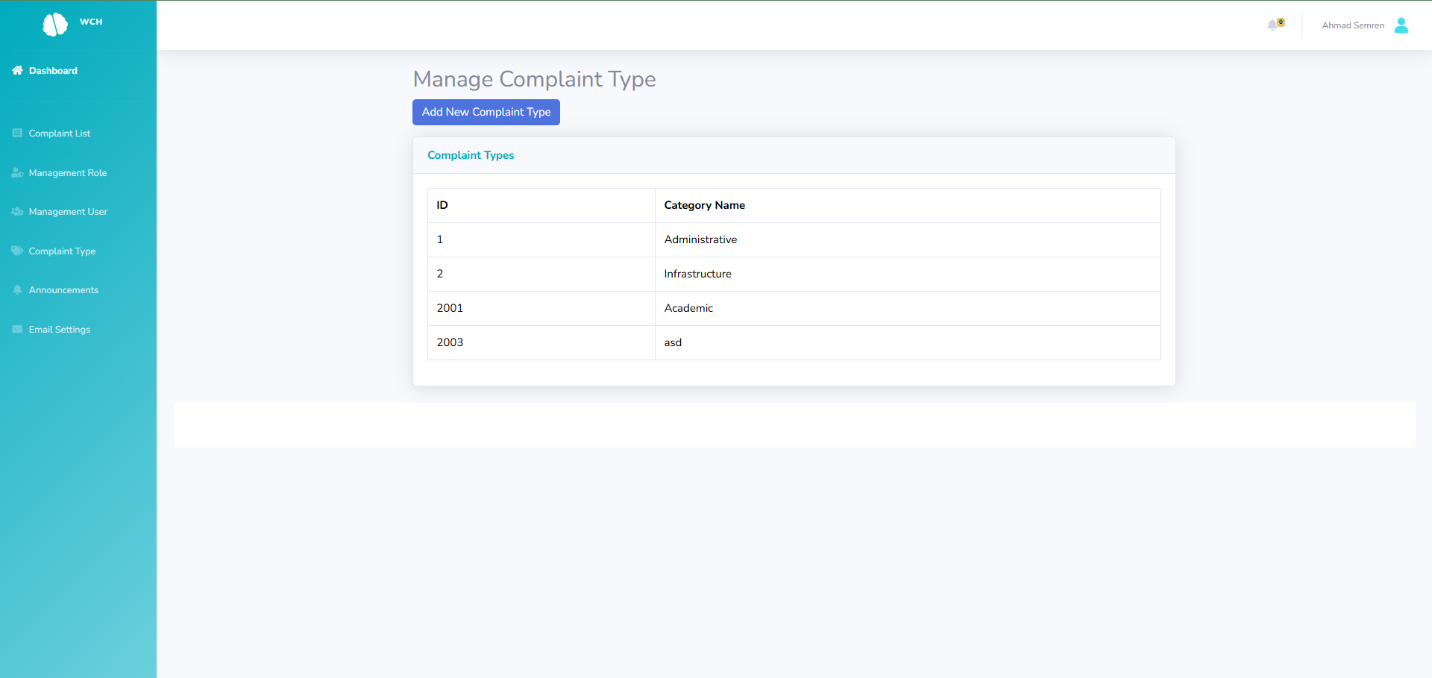


Figure manage complaint type

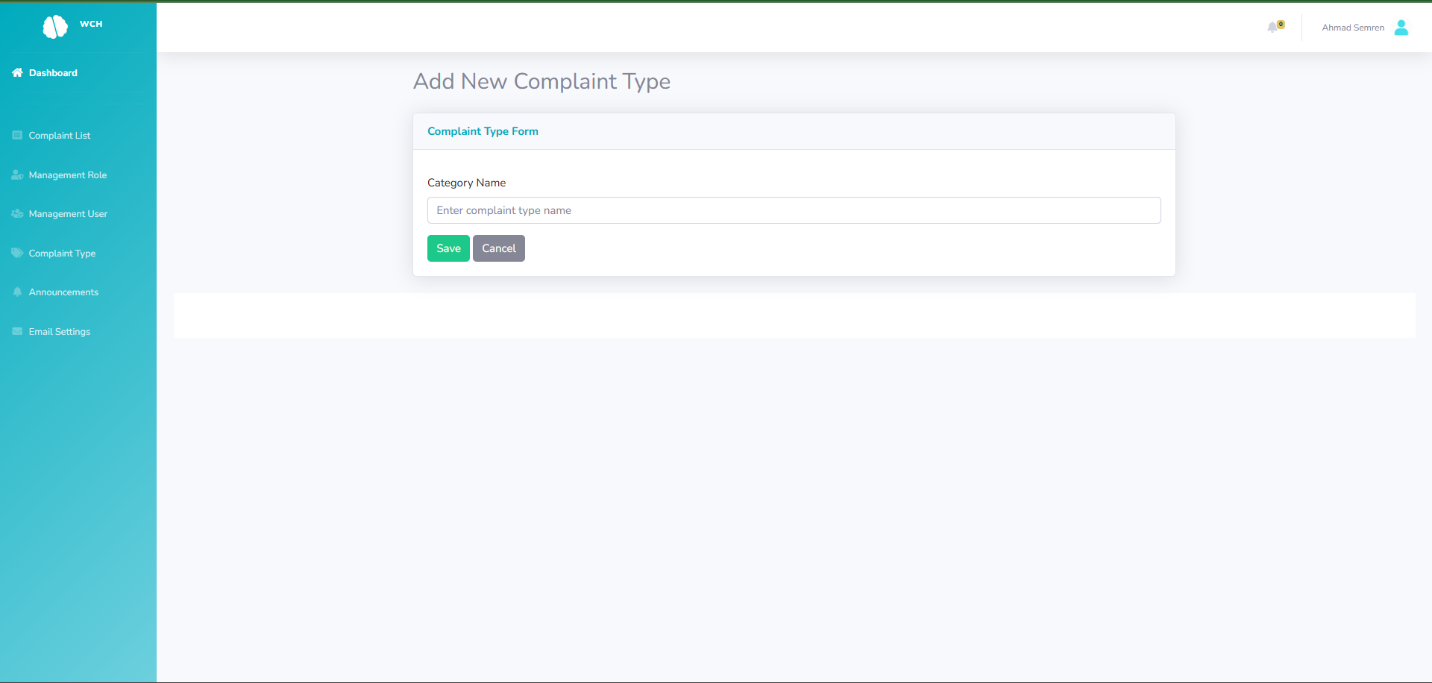
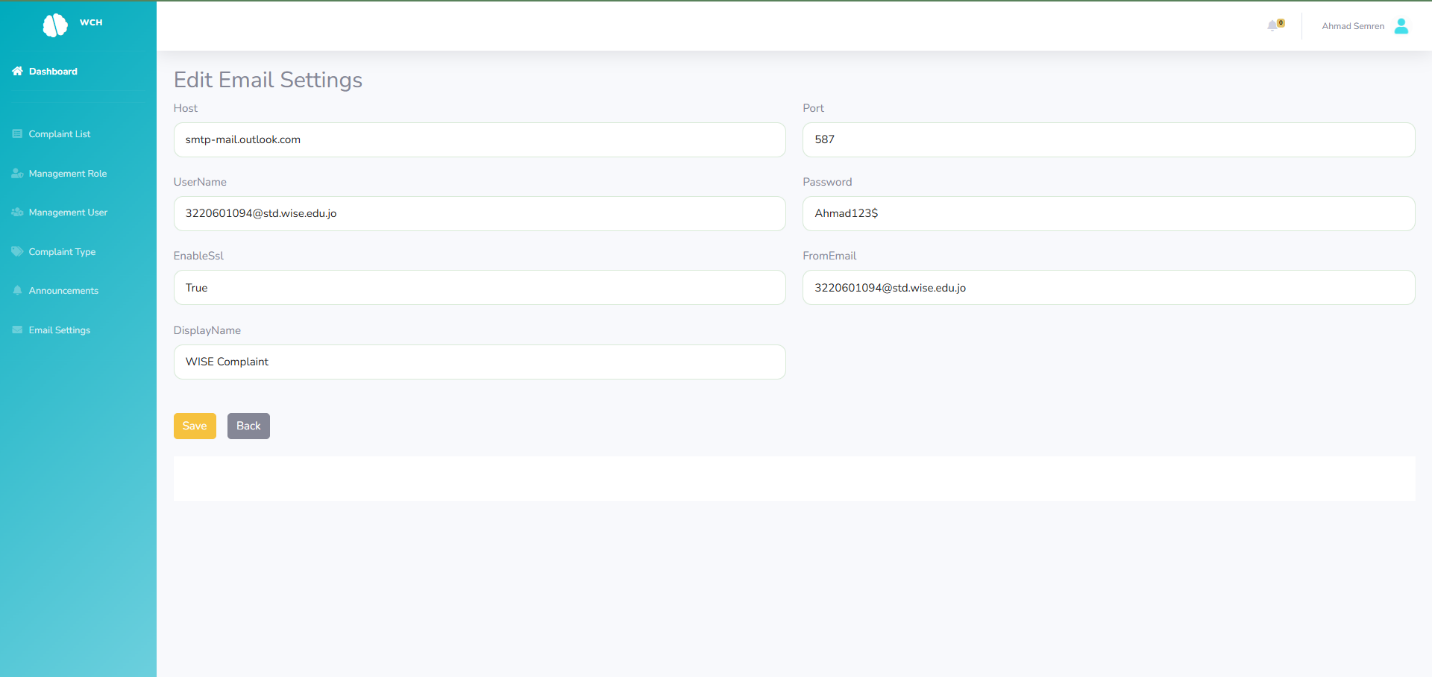


Figure edit email setting

Figure add new complaint

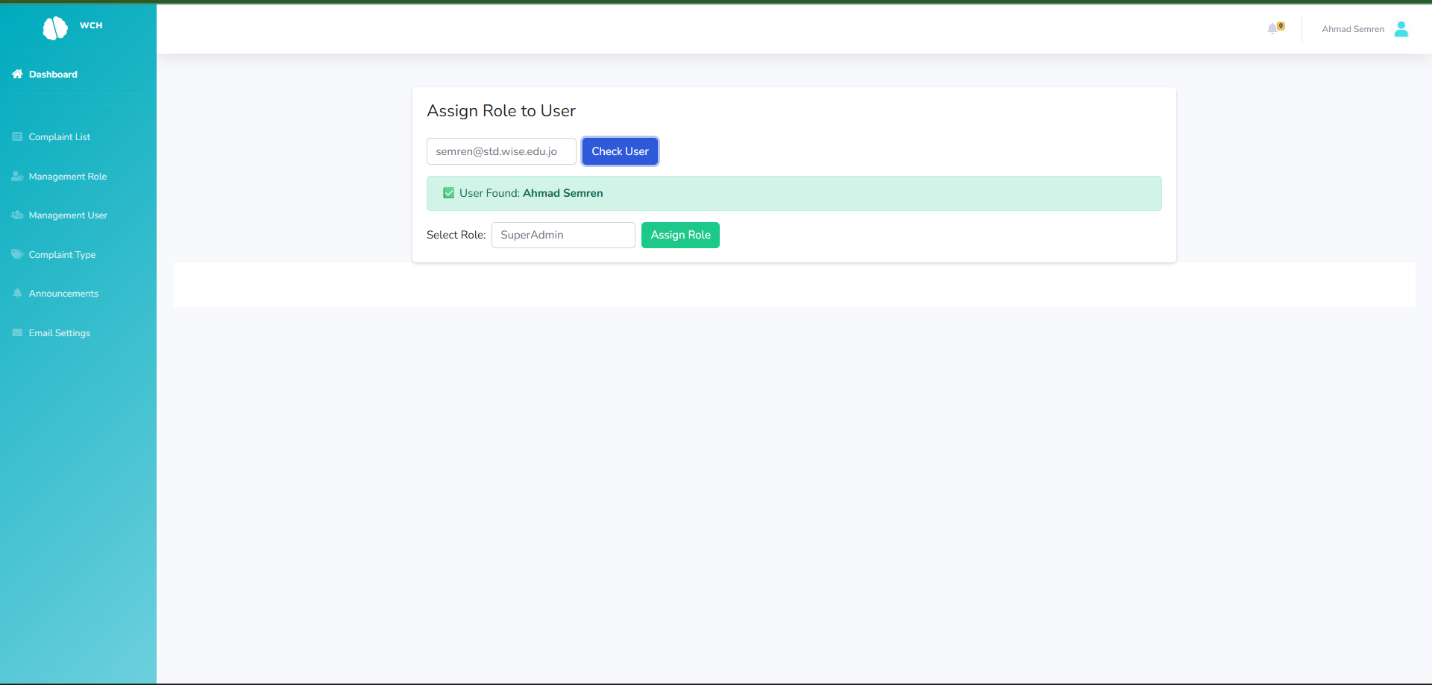
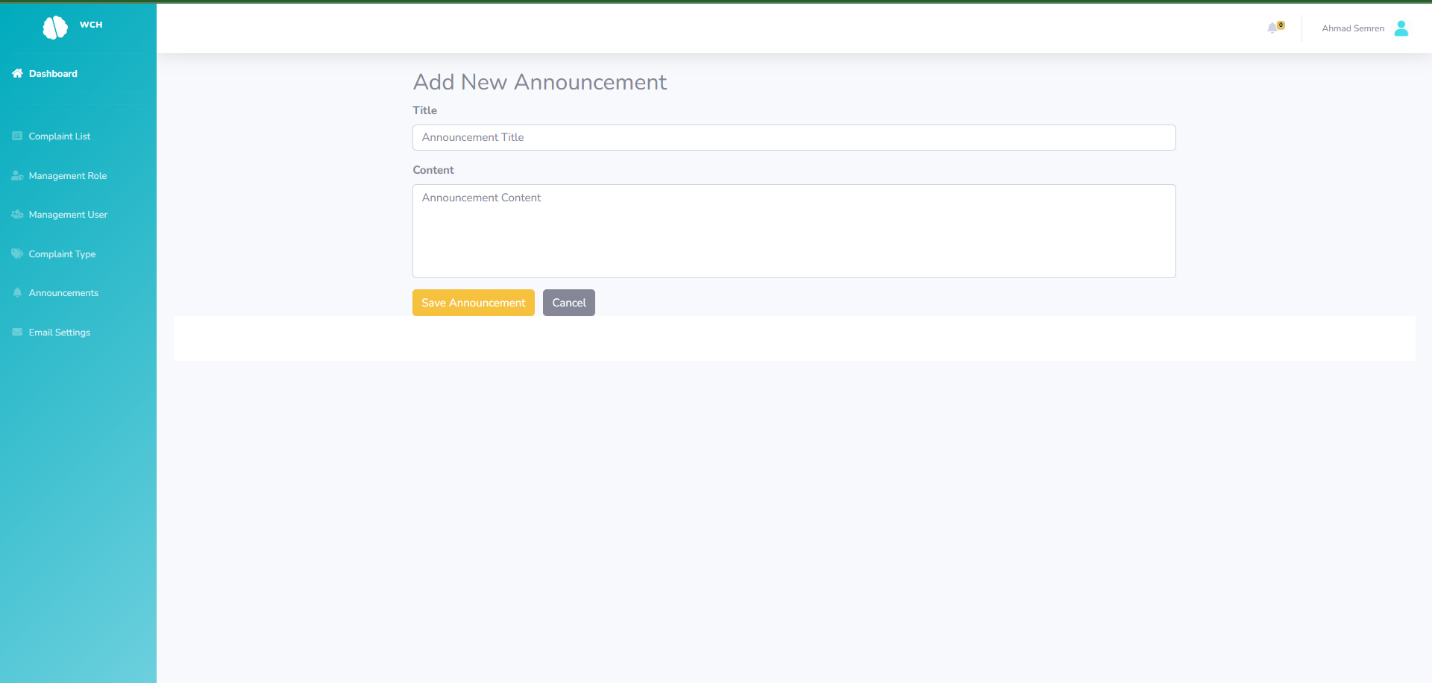


Figure add new announcement

Figure assign role to user