**CHAPTER ONE**

**INTRODUCTION**

This chapter describes the project and provides some background information. Furthermore, this chapter follows the project's problem statements to provide a clear understanding of the project's scope and objectives. This chapter will serve as a roadmap for all subsequent stages of development.

**1.1 Background of The Study**

An examination is defined as the assessment of a person's understanding of their knowledge. A formal test may be used to conduct the evaluation. The exam is taken to evaluate a student's proficiency in a particular subject. Examinations can take many forms, including written tests, oral exams, practical demonstrations, or a combination of these methods. The purpose of an examination is to determine a person's level of knowledge or skill in a particular area and to provide an objective measure of their abilities. As a student in higher institutions, the importance of exam preparation cannot be overemphasized because it helps you to gain a better understanding of the material you have been studying. (teachmint, 2022).

Surbhi (2018) defines management as the practice of managing and regulating the affairs of an organization, regardless of its nature, kind, structure, or size. It is the act of building and sustaining an atmosphere in which people of a company may collaborate and achieve business objectives efficiently and effectively. We may define Complaint Management as the process of regulating concerns of discontent or displeasure voiced by complainants. Complaint management allows for the effective processing and control of concerns raised by complainants. Management serves as a guide for a collection of individuals working in an organization, directing their efforts toward a single goal (Surbhi, 2018).

According to Advocacy Research (2021), it is critical to file concerns before they can be managed. When you have a complaint, it is critical to gather information and offer evidence when possible.

A good Complaint Management System is essential for offering good complaint service. It aids in the measurement of complaint satisfaction and serves as a valuable source of information and feedback for service improvement. Complainants are frequently the first to notice when something isn't operating properly (Ombudsman, 2020).

This system would aid in reducing the problems that students face when filing complaints about their examination results. There are various advantages to filing complaints. They assist the administrator in identifying critical areas for service development, keeping top management aware, improving service, and, most importantly, improving complaint communication.

The development and implementation of a web-based student examination result complaint system will aid in the maintenance of an effective, fast, and equitable complaint-handling system that is easily accessible and free of charge to students.

**1.2 Statement of the Problem**

Examination results are an important aspect of a student's academic career, as they reflect the student's knowledge and understanding of the subject matter. However, there may be instances where a student believes that their examination results are not an accurate reflection of their abilities, or that there has been some kind of error or mistake in the grading process. In such cases, students must have a way to voice their concerns and seek resolution. The system should provide an interface for students to enter their complaint and any supporting evidence, and should also allow them to track the status of their complaint as it is being processed.

**1.3 Aim and Objectives of the Study**

To develop a web-based student examination result complaint system for the department of computer science at Kaduna polytechnic.

**Objectives**

The objectives of this research work are as follows:

1. To design a working platform where **that allows students to easily and securely submit complaints about their examination results**.
2. Unit and integration testing will be performed to validate the design's effectiveness and efficiency, as well as to guarantee that the functionalities are error-free.
3. To evaluate the existing paper-based information management system

**1.4 Scope of the Study**

The study will not address issues related to the validity of examination results or the handling of complaints by the relevant authorities. The focus of the study is solely on the development of a web-based student examination result complaint system that allows students to submit complaints about their examination results, and track the status of their complaint as it is being processed.

**1.5 Limitations of the Study**

This study's scope has been constrained by several core issues, including:

**Time** - The researcher's everyday busy academic pursuits limited the time allotted for research for this study.

**Access to literature** – Access to some material was restricted, although the available material was optimized.

**1.6 Significance of Study**

The development of a web-based student examination result complaint system will provide a valuable resource for students to voice their concerns and seek resolution in cases where they believe their examination results are not accurate. The system will also provide a more efficient and streamlined process for the relevant authorities to handle student complaints which could potentially improve the speed and fairness of the complaint process, as well as increase the accountability of the educational institution.

**1.7 Project Organization**

The project is divided into five chapters. The outlines are presented below:

**Chapter One: Introduction**

Chapter one introduces this project work, the study's background, the problem statement, the purpose and objectives, the scope of the study, the constraints of the study, the relevance of the study, the project organization, and the definition of terms.

**Chapter Two: Literature review**

This chapter focuses on the literature review, and the contributions of other scholars on the subject matter being discussed.

**Chapter Three: Methodology and Design**

This chapter is concerned with the presentation of the results of system analysis and design. It presents the research methodology used in the development of the system to facilitate an understanding and effective future implementation of the system.

**Chapter Four: System Implementation Evaluation**

This chapter describes the system implementation and documentation, analysis of modules, and system requirements for implementation.

**Chapter Five: Summary, Conclusion, and Recommendation**

The chapter provides a summary of major findings, conclusions, and recommendations based on the study conducted.

**1.8 Definition of Terms**

1. **Web-based**: This term refers to a system that is accessed and operated through a web browser, rather than through a standalone program or application.
2. **Student**: In this context, a student is someone who is enrolled in a school or educational institution and is taking classes or exams.
3. **Examination**: An examination is a formal test or assessment of a student's knowledge or skills in a particular subject or course.
4. **Result**: The outcome or score of an examination, indicating how well a student performed.
5. **Complaint**: In this context, a complaint refers to an expression of dissatisfaction or disappointment with the results of an examination.
6. **System**: A system is a set of interconnected components or elements that work together to achieve a specific goal or function. In this case, the system is a web-based platform that allows students to file complaints about their examination results
7. **User interface** **(UI)**: The part of a software application that the user interacts with, including the layout, buttons, and other elements

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

The purpose of this chapter is to show how the problem under consideration relates to prior research, current practice, or other fields of knowledge by citing relevant works by other researchers who have dealt with a similar issue. Furthermore, this chapter will include a synthesis of current research on the issue, highlighting areas of agreement, disagreement, and gaps in the literature, to establish the significance of the project topic in the field and to recommend opportunities for future study.

**2.2 Literature Review**

Anusiuba et al. (2021). Design and Implementation of a Tertiary Institution Web-Based Student Complaint Management System. In comparison to a computerized system, certain complying processes are highly demanding, inefficient, and require needless personnel. Most universities' complaint systems are now manual, and even if they are automated, the response time to complaints is longer. The procedure works as follows: regardless of the type of complaint, the student sends a formal letter to the Head of the Department, detailing the situation in detail. The letter is then forwarded by the Head of the Department to whoever is in charge of the complaints. These processes appear simple, but they are not since the entities are not entirely responsible for complaints, thus it may take some time for the letters to be transferred to the proper channel.

Moreso, the Waterfall Approach was used for this study's research methodology. The data flow was represented in the system design to capture the techniques employed in the system. PHP programming language, JavaScript, Hyper Text Mark-up Language (HTML), Cascading Style Sheet (CSS), and MySQL are the tools utilized for development. The Database Management System (DBMS) was provided by the MySQL server program, while the Integrated Development Environment (IDE) was provided by the Visual Studio Code application (IDE).

In conclusion, a student can use the complaint management system to make a complaint, monitor the progress of their complaint, and still obtain comments from the administrator. When detailed complaints are kept in a database, they are easier to retrieve at any time. The right complaint system provides the school with a new degree of control over operations, increases efficiency, and aids in the proper expression of grievances. Switching from a manual to a computerized complaint system can be tough; nevertheless, the advantages outweigh the obstacles, and it can be well worth the time and effort.

Singh et al. (2021). Online Complaint Management System.The system offers an online solution to the public's concerns, saving time and eradicating corruption. The goal of the complaint management system is to make it simpler to coordinate, monitor, track, and handle complaints by recording the status of complaints made by the public to the department.

Furthermore, it will be Dynamic and Fast Data. All customer/recipient information will be carefully maintained. The assignment of complaints to various staff shall be done correctly in order to avoid duplication. It will build a platform where no customer or complaint record will ever be lost. Within one to two days, the complaint will receive an automatic response and answer viewing. If any employee fails to do their assigned tasks, severe consequences will be imposed.

In conclusion, application software has been developed successfully and was also tested successfully by taking "test cases". It is user-friendly and has the necessary choices that the user may use to complete the stated actions. To a large extent, application software satisfies the defined knowledge criteria. The system was created with these and future requirements in mind, and it is extremely adaptable.

Ema et al. (2022). Development of a Web-based Water Supply Complaint Management System. The Internet's broad use as a communication medium today has the ability to help bridge the gap between the people and the government. Using an adequate information system may provide users with additional benefits such as the ability to identify and categorize each complaint based on its severity level, the eradication of false complaints, and the ability to accept complaints outside of usual business hours. Following that, an online complaint, also known as an online complaint management system (OCMS), is a web-based method for handling public complaints while saving time and decreasing corruption.

Furthermore, the method was also meant to assist local governments or water supply distributors in effectively managing complaints and acting fast. After multiple functional tests, user acceptability testing is conducted to assess the proposed system. The System Development Life Cycle (SDLC) was used to design the system, and the waterfall model was chosen as the technique. Thirty people from Sibu, Sarawak, were chosen to participate in the testing phase and were requested to fill out questionnaires separated into four categories: user interface satisfaction, ease of use, usability, and function.

In conclusion, the Water Supply Complaint Management System was able to help customers by simply filing valid complaints about water supply problems in the neighborhood. Furthermore, by entering into the system or receiving an SMS message, consumers may check all submitted complaints and their status. SMS notification is seen as critical in locations with limited Internet access. Our everyday lives are dependent on having access to clean water thus it is vital that local governments act quickly when a water crisis emerges.

Kormpho et al. (2018). Smart Complaint Management System. Nagpur Municipal Corporation is responsible for providing Nagpur's people with essential urban services, as a result, NMC is in charge of administration and basic infrastructure for the city. It takes a long time to file a lodging complaint in NMC, and the victim must travel to the municipality office and wait in line. The "Smart Complaint" app is introduced to give users a platform to conveniently make a complaint. As a result, “Smart Complaint” lowers people's efforts. The complaining lodger might reveal their whereabouts by utilizing GPS. This app is concerned with the internal handling of complaints. The primary goal of this system is to assist the public in learning about their location and resolving concerns online without having to visit the office on a frequent basis until the problem is resolved.

Moreso, the system was created and implemented as a mobile application enabling citizens to file complaints about problems that may be resolved by municipal corporations. Because people currently utilize Android phones, they designed an app that can be used to register a complaint in a very easy method by uploading a photo of the suspected area and using GPS to report the position of that spot to corresponding Municipal Corporations. The system also includes a web platform that allows residents to submit complaints and also assists Municipal Corporation employees in easily resolving complaints by streamlining the process of passing complaints to sub-officers.

Rajeswari & Nur (2019). Mobile Complaint Management System (MOCAS). With the increase in student enrollment, a more comprehensive monitoring system is required, as managing manually becomes inefficient. Students must go to the college administration to fill out a complaint form for the institution's facilities problems. Filling out the form takes a long time. Not only are the forms filled out manually, but all of the complaint forms are maintained in a file with stacks of paper, containing information about the facility's state, which is documented by the personnel after they have completed the repairs.

Furthermore, the chance of the complaints information being duplicated exists, making it time consuming for the employees to identify the correct details of the complaints. In the existing method, complaints were not stacked up according to the sort of harm. This makes it difficult to obtain specific complaint data for use as a reference or proof in some circumstances. It is pretty difficult to solve and sometimes takes longer than the desired time period.

In conclusion, MOCAS saves the complainant time by answering the complaint using a mobile application. The complainant does not need to visit the college office regularly to file a complaint while going about their daily activities. MOCAS also supports them in presenting a status for each complaint that has been resolved by personnel. As a result, the complainant does not need to go back and forth to determine whether or not the issue has been resolved. Several factors must be addressed while developing a mobile application, including dependability, availability, maintainability, and safety. Furthermore, the system must take into account accuracy, security, responsiveness, and testability.

Sudhir et al. (2020). Electronic Complaint Management System for Municipal Corporation. The major goal of the project is to use this internet application to assist the people who are experiencing various challenges in their communities. This project has the ability to bridge the gap between citizens and the government. It can control unethical bribery work and even shorten processing time.

Moreso, the basic concept of this project is the detection and resolution of people's complaints, as well as their correction within the system-generated time restriction. The system generates a clear report that includes the assignee's name, complaint type/department, and so on. All of the aforementioned characteristics are useful when examining a complaint report.

In conclusion, the adoption of this project will provide exceptional results, and it will also aid in stimulating the creation of similar or more complicated complaint management systems. In general, complaints and other forms of feedback play a vital part in the growth of any organisation and in improving interactions with consumers. This system may be viewed as a precursor to the systems that will be created in the future in relation to complaint management systems.

**2.3 Summary of Related Literature Reviews**

|  |  |  |
| --- | --- | --- |
| **Author & Year** | **Title & Description** | **Merit and Demerits** |
| Anusiuba et al. (2021). | Design and Implementation of a Tertiary Institution Web-Based Student Complaint Management System.  The system will aid in resolving some of the issues that students face in the academic setting. | The program is both effective and efficient in accomplishing project objectives while also being user-friendly.  The system is limited only to the web. |
| Singh et al. (2021). | Online Complaint Management System.  The system offers an online solution to public concerns, saving time and eradicating corruption. | This system makes the filling of complaints to be easy and well-coordinated.  The system might lack scalability. |
| Ema et al. (2022). | Development of a Web-based Water Supply Complaint Management System.  This paper aims to present a technology-based online notice board using the Internet of Things (IoT) for the dissemination of information that is cost-effective. | The system helped in assisting the government with managing complaints.  The system lacks good user interface. |
| Kormpho et al. (2018). | Smart Complaint Management System.  The research created an Android-based information notification application with push messaging services to distribute information from the school to students or parents via push notification, and after tapping the notice, the specifics of this information may be viewed. | The location of the user can be tracked easily with the help  of a GPS system and the system is user-friendly.  The system is limited to just android users. |
| Rajeswari & Nur (2019). | Mobile Complaint Management System (MOCAS)  Water Supply Complaint Management System was able to help customers in making legitimate complaints about water supply problems in the community in a timely and easy manner | The system integrated push notification  The application is limited to android users |
| Sudhir et al. (2020). | Electronic Complaint Management System for Municipal Corporation.  The main purpose of the project is to help the public who are  facing different problems in the localities by this online  application | The system eradicated the manual book keeping complaint system  Poor user experience |

**2.4 Analysis of the Current System**

The current system manually registers and manages complaints. In the current system, the processes used to handle various complaints vary. This study focuses on the complaints indicated in the scope.

When there are complaints about missing scripts, the current system goes through a number of processes.

1. The complainant (student) writes a formal letter to the Head of the Department outlining the difficulties.
2. The letter is forwarded by the Head of the Department to the department in charge of outcomes.
3. The results department transmits the letter to the ICT officer in charge of uploading results.
4. The in-charge ICT officer investigates why the result was missing and attempts to fix it

**2.4.1 Problem Inherent in** **the Current System**

There are several problems inherent in the current system of filling examination results complains they include:

1. Inefficiency: manually processing complaints can be time-consuming and error-prone, leading to delays in resolving issues.
2. Lack of transparency: without a clear record-keeping system, it can be difficult to track the progress of complaints and ensure that they are being handled fairly.
3. Limited scalability: manual systems can be overwhelmed by a large number of complaints, making it difficult to handle a high volume of complaints.
4. Inadequate record-keeping: with the manual system it is difficult to maintain accurate records of complaints, which can make it difficult to identify patterns or trends.

**2.5 Analysis of the New Proposed System**

The study's proposed system is a web-based examination result complaint system that would save the time of students by directly launching complaints with the help of the proposed system. Students don’t need to go to the department for launching complaints. The proposed system will improve efficiency by reducing the time consumed in handling students’ complaints

**2.5.1 Advantages of the Proposed System**

1. In today's technology environment, this technique allows anyone to make a complaint using a smartphone application, which saves them time.
2. It will save time and effort by filing the complaint online rather than manually.
3. Also, the status of the complaint, that is whether it is denied, accepted, processed, or resolved, can be easily followed.
4. It is user-friendly and cost-effective.

**CHAPTER THREE**

**METHODOLOGY AND DESIGN**

**3.1 Introduction**

A methodology is a rigorous study or inquiry, particularly to unearth new facts or information; thus, research methodology should be good enough to enable the achievement of the specified objectives, which are achievable using specific components, such as data collection and design procedures, and system modeling (use case, activity, and class diagrams). This chapter provides the input/output specifications as well as the system requirements for developing a web-based student examination result complaint system.

**3.2 Methods of Data Collection**

Before constructing any system, it is necessary to collect data and facts about the existing system to comprehend what is going on. Two approaches were used in this study.

1. Primary Source
2. Secondary Source

**3.2.1 Primary Source of Information**

This includes data gathered directly or indirectly from target users, with no edits or suggestions from other writers. This main source's material is considered more accurate and credible. As a result, the goal is to incorporate the knowledge gleaned from this source into the project to satisfy the criteria. Interviews and observations were used as primary source data collection strategies.

**3.2.2 Secondary Source of Information**

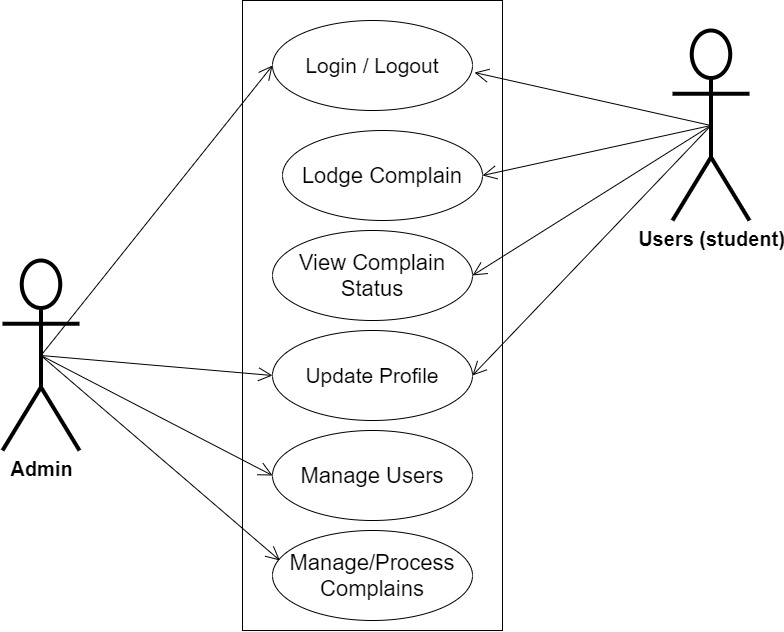
This essentially includes all of the information that someone can receive from existing sources such as books, the internet, case studies, articles, newsletters, and other relevant publications. The resources obtained from the internet in particular were quite relevant; various search engines, particularly Google, made it very easy to find information.

**3.3 System Modeling**

A system model is a conceptual model of a system that explains and depicts it. A system is any interaction between a group of components that work together to achieve a common purpose. A collection of visual notation techniques inherent in the Unified Modeling Language, which was used to design this current system, may be used to construct visual models of object-oriented software-intensive systems. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

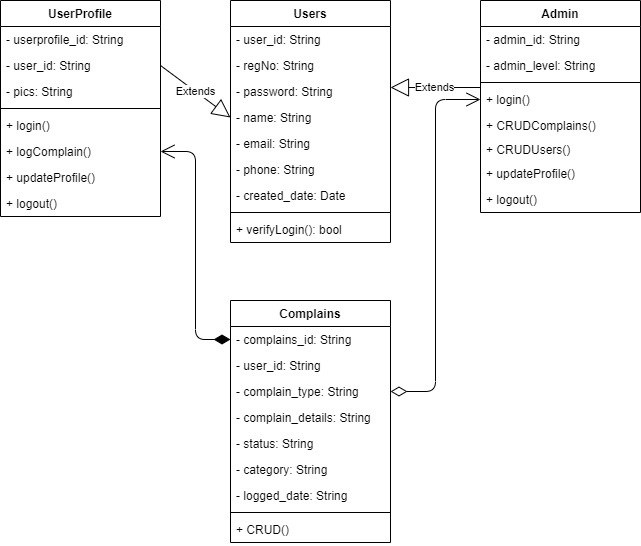
Use cases are groups of interactions between a system and a user. Use case diagrams are used to graphically portray the functioning of a system in terms of its actors, goals (expressed as use cases), and dependencies between those use cases.



**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagrams**

The Unified Modeling Language (UML) class diagram is an implementation of an independent view of how the system interface might appear, with each class having its own set of properties and displaying how they interact with one another. Class diagrams use the Unified Modeling Language standards to visually depict a given system's static structure and composition (UML).



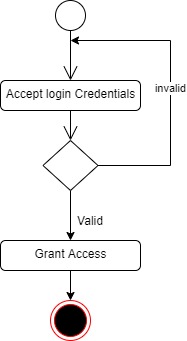
**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

An activity diagram, like a flowchart or a data flow diagram, visually illustrates a series of events or the flow of control in a system, but it acts more like an enhanced version of both.

**Login**

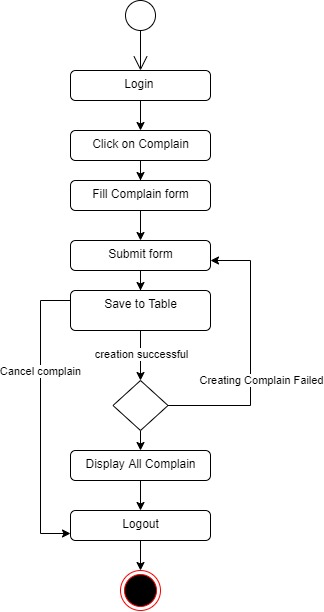
The process for gaining access to the system is depicted in the diagram below; the email address and password must be accurate to gain access.



**Fig 3.3.1 Login Activity Diagram**

**Lodging Complain**

The process for filing a complaint is depicted below, to create a complaint one has to be authenticated to lodge a complain.



**Fig 3.3.2 Lodging Complain Activity Diagram**

**3.4 Database Design**

Input specification is the logical explanation of how data is stored in the computer's memory. SQL standards are vital for guaranteeing that structured data is uniform and independent of applications due to the flexibility experienced when using the system, as well as the simplicity of accessing and reading the data and ensuring applicability throughout the internet. The following are some of the input specifications used in this project effort.

1. Users Table: contains basic information about all system users.
2. Complain Table: contains every system-saved complaint information.

**Table 3.1 Users Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| user\_id | Varchar | No | PK | 32 | Unique string for identifying users |
| regNo | Varchar | No |  | 14 | User Registration number |
| password | Varchar | No |  | 128 | User Password |
| name | Varchar | No |  | 100 | User full name |
| email | Varchar | No |  | 100 | User email address |
| phone | Varchar | No |  | 11 | User phone number |
| created\_date | Varchar | No |  | 10 | Account created date |

**Table 3.2 Complain Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| complain\_id | Varchar | No | PK | 32 | Unique string identifying notice |
| user\_id | Varchar | No | FK | 32 | Reference to the user account |
| complain\_type | Varchar | No |  | 255 | Type of the complaint |
| complain\_details | Varchar | No |  | 255 | Detail description of the complain |
| status | Varchar | No |  | 20 | Status of the complaint |
| category | Double | No |  | 20 | Category of the complaint |
| logged\_date | Date | No |  | 10 | Date the complaint was logged |

**3.5 Output Design**

This declares and displays the outcome of the given input. This automated system's output is dependent on its input. The output specification is listed below.

**Table 3.3 Users** **output design table**

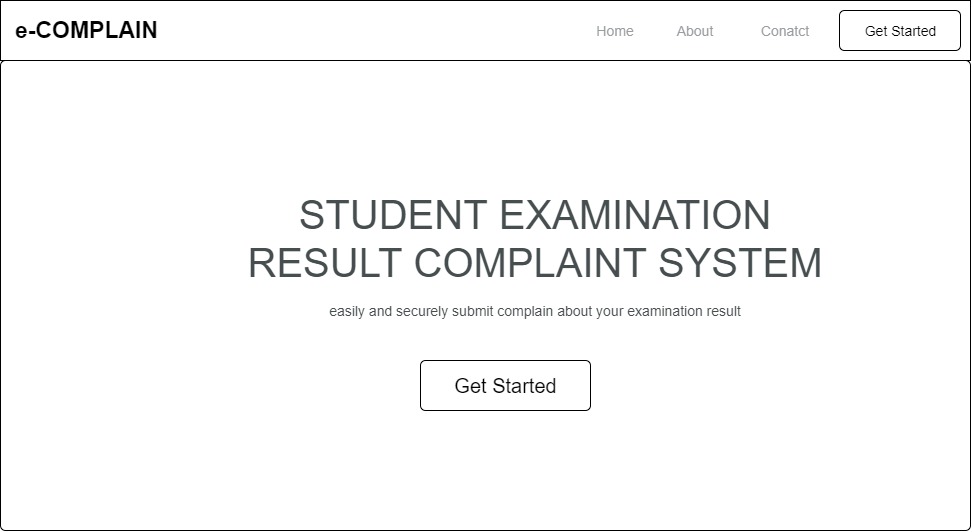
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User\_id** | **regNo** | **Name** | **Email** | **Phone** | **Created\_date** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.4 Complain** **output design table**

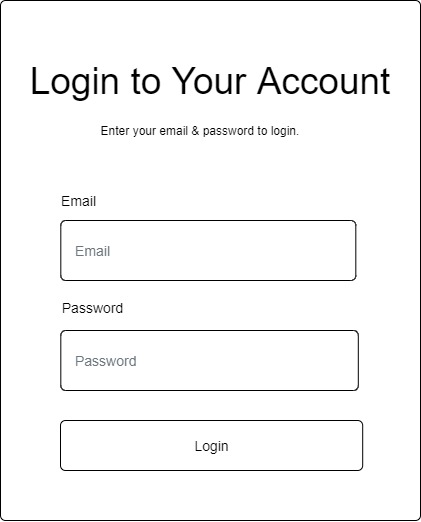
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Complain\_id** | **User\_id** | **Complain\_type** | **Complain\_details** | **Status** | **Category** | **Logged\_date** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**3.6 Input & User Interface Design**

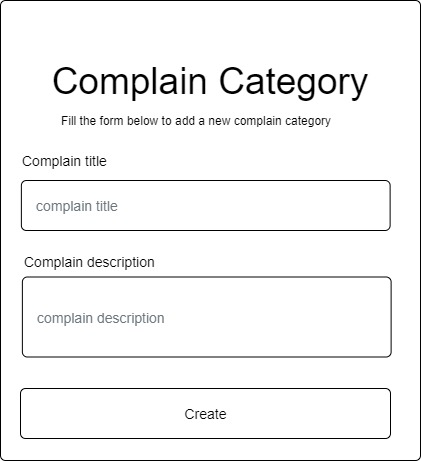
This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be fully secured, thus authentication will be required to see various levels of the information. To help with the designs, a mid-fidelity wireframing program called Draw.io is employed.



**Fig 3.6.1 Home Page**



**Fig 3.6.2 Login Page**



**Fig 3.6.3 Create Complain Category Form**

**3.7 System Requirement**

Every piece of software-generated has predefined system requirements that it must fulfill to function properly. The system requirements, on the other hand, are the bare minimum of hardware and software required for the system's intended operation.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 2 GB of RAM (Random Access Memory).
2. Minimum of Intel Dual core processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 7 OS (Operating System).
2. Vs. Code IDE installation.
3. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This study will be a web-based application based on the design of a relational database (SQLite). HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript will be utilized for front-end development, while Django (Python) will be used for back-end programming.

**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION EVALUATION**

**4.1 Introduction**

The implementation process of the new system is thoroughly described in this section, emphasizing its efficiency and effectiveness. It provides real-life examples of how the system functions and outlines the sequential steps required for its successful implementation.

* 1. **System Testing and Evaluation**

Conducting thorough testing of the developed system is essential for multiple reasons. One significant objective is to identify and address any potential shortcomings or defects in the system. In this project, a blend of unit and integration testing was utilized to validate the design's efficacy and efficiency, guaranteeing that the new system meets its functional requirements and operates without any errors.

**Unit Testing**

This part examines specific units or single components of the system individually to confirm that specific phases function properly and without problems.

**Integration Testing**

Integration testing was performed on the software, wherein all components were brought together and operated as a unified system. The objective of this testing was to validate the connectivity and proper integration of the various parts, ensuring seamless collaboration among the units.

**4.3 System Installation**

In order to use the proposed application on any computer system, the following steps need to be taken:

1. Make sure, pip, pipenv, and python3 or greater are installed on the system.
2. Copy your project folder to any location of your choice.
3. Open project folder in Visual Studio Code
4. On the terminal run “pipenv install -r requirements.txt”
5. On the terminal run “python manage.py runserver”
6. Open any browser on the system example Chrome, Microsoft Edge, or Mozilla Firefox.
7. On the address bar, type <http://127.0.0.1> and press the enter key the site should be loaded.

**4.4 Security Measures**

Given that the website has a public scope, certain information such as the index page and login page are accessible to all visitors. However, there are specific functionalities and information that are restricted and not available to every visitor. To enforce these restrictions, passwords are implemented, granting different levels of access to users. The administrator holds the highest level of access, followed by users with lesser privileges. This approach ensures that certain features and data are only accessible to authorized individuals while maintaining the overall public availability of the website.

**4.5 Sample Outputs**

These describe and give the pictorial representation of the program or software; it shows and gives a clear understanding of the design, and displays all the interfaces.

**Homepage**

The page serves as a gateway to navigate and explore the various sections of the website

Fig 4.1 Homepage

**User Login**

This is a page that grants users (admin, staff, and student) access to the system only if the correct credentials are provided

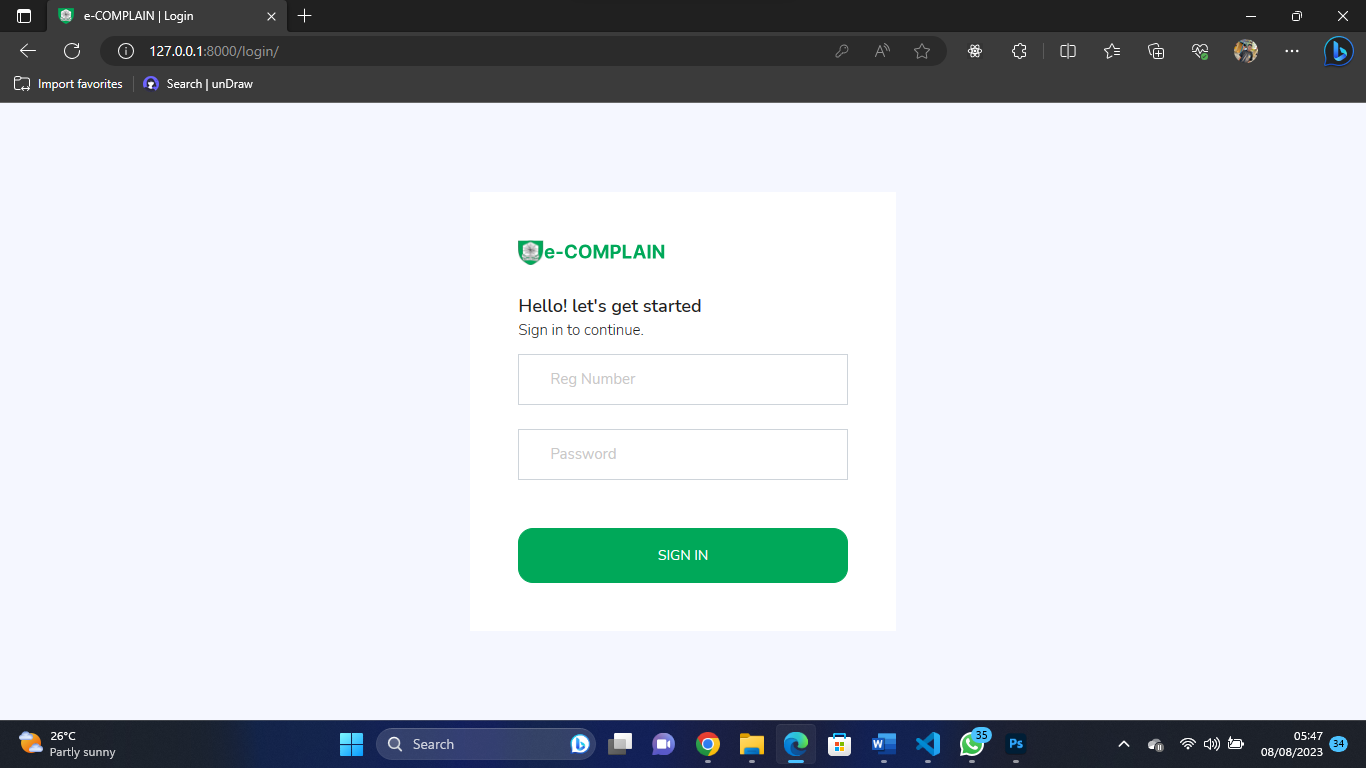


Fig 4.2 User Login

**Admin Dashboard**

This is the clearance admin dashboard, the sidebar shows the available functionality for the admin

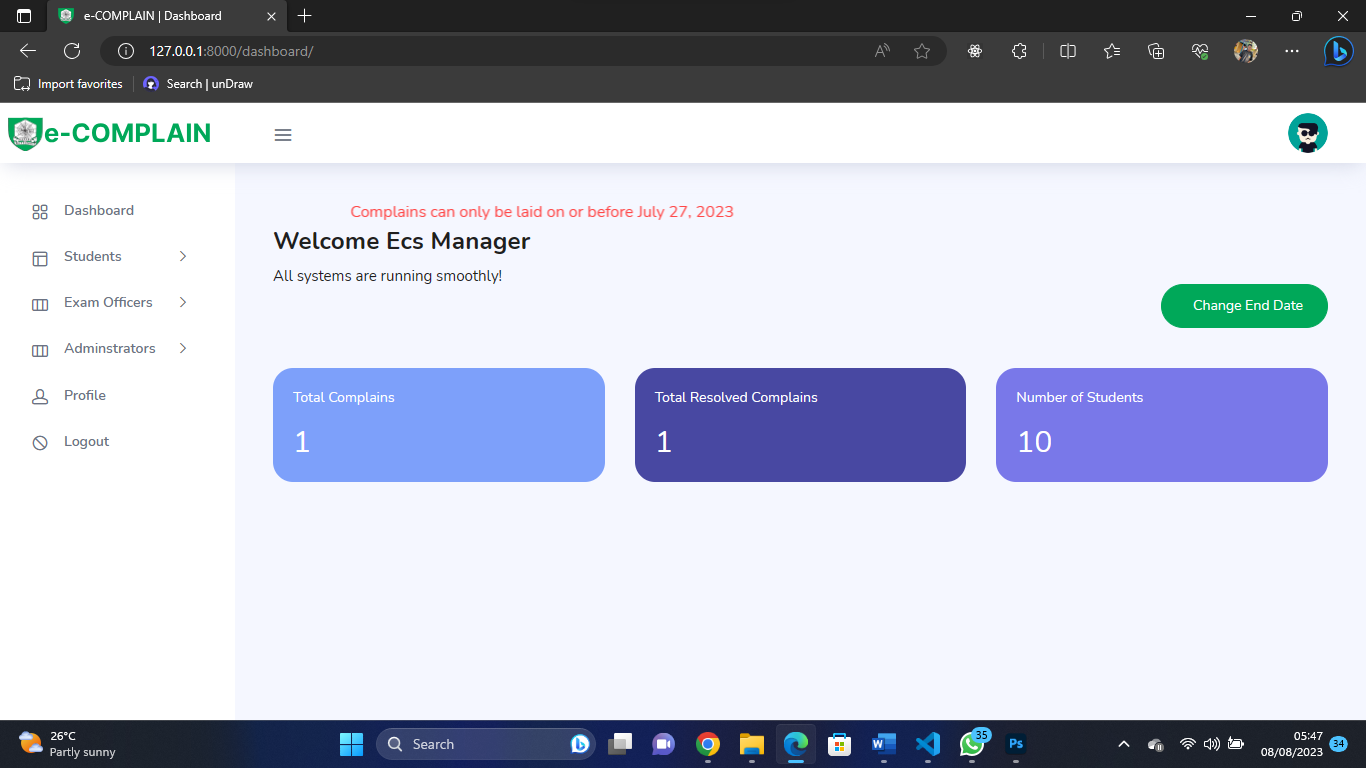


Fig 4.3 Admin Dashboard

**Manage Student’s Account**

This is the page where the admin can effectively create and manage each student account

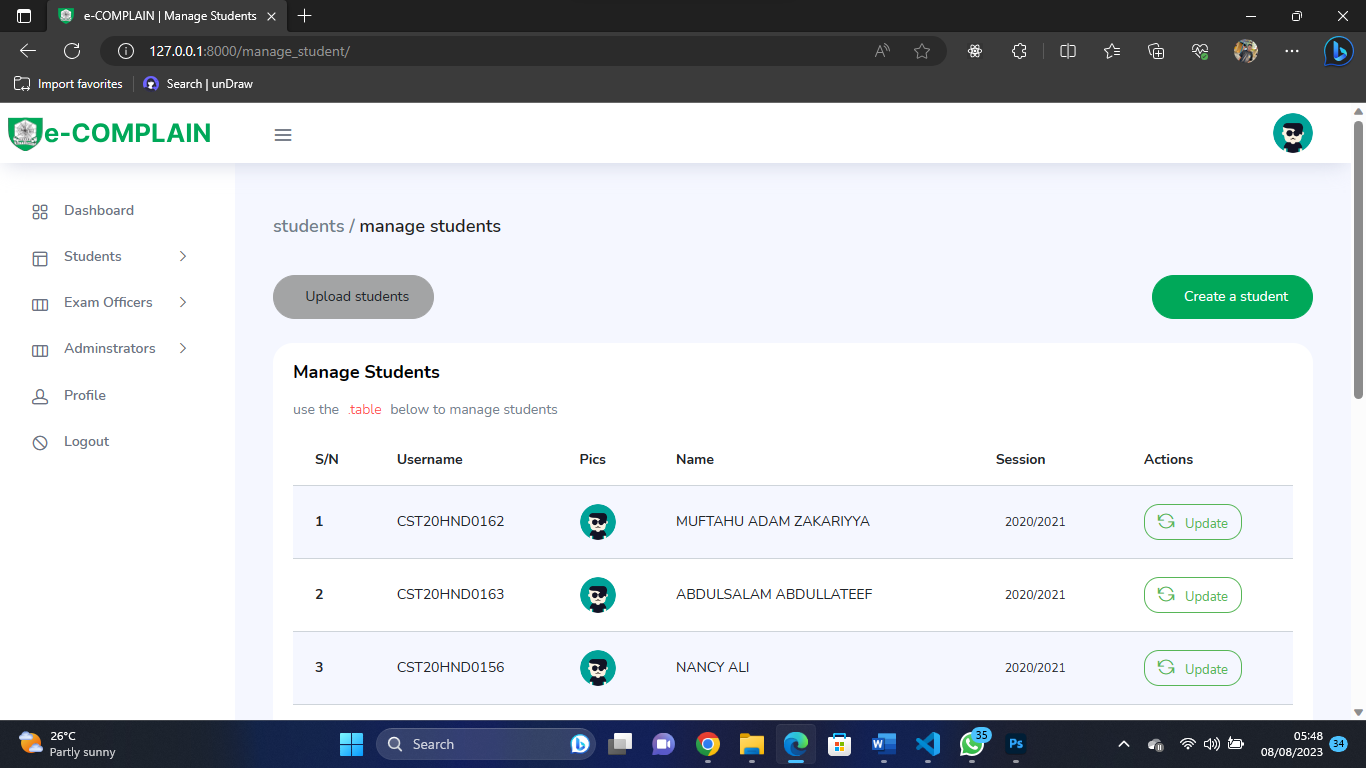


Fig 4.4 Manage Student’s Account

**Manage Exam Officers**

This is the page where the admin can effectively create and manage each exam officers account

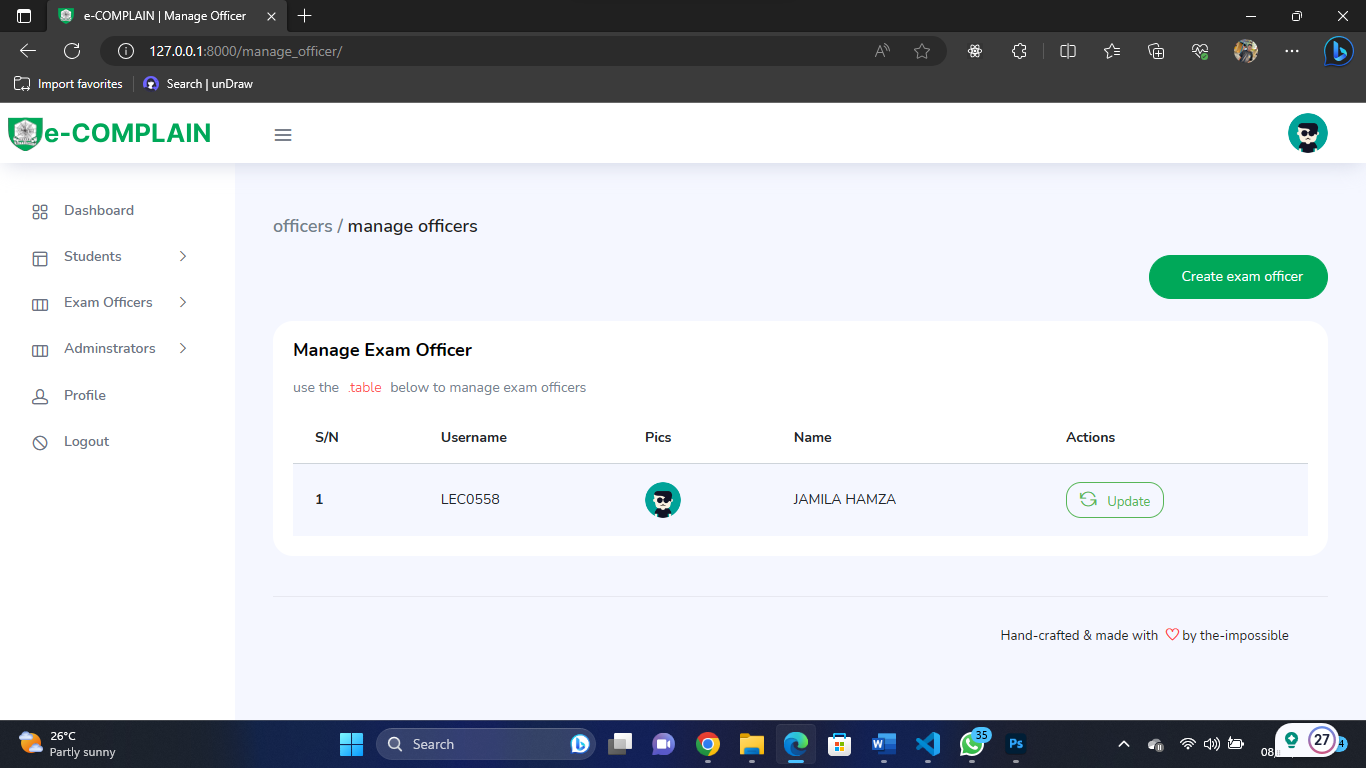


Fig 4.5 Manage Exam Officers

**Manage Administrative Offices**

This is the page where the admin can effectively create and manage each administrative account

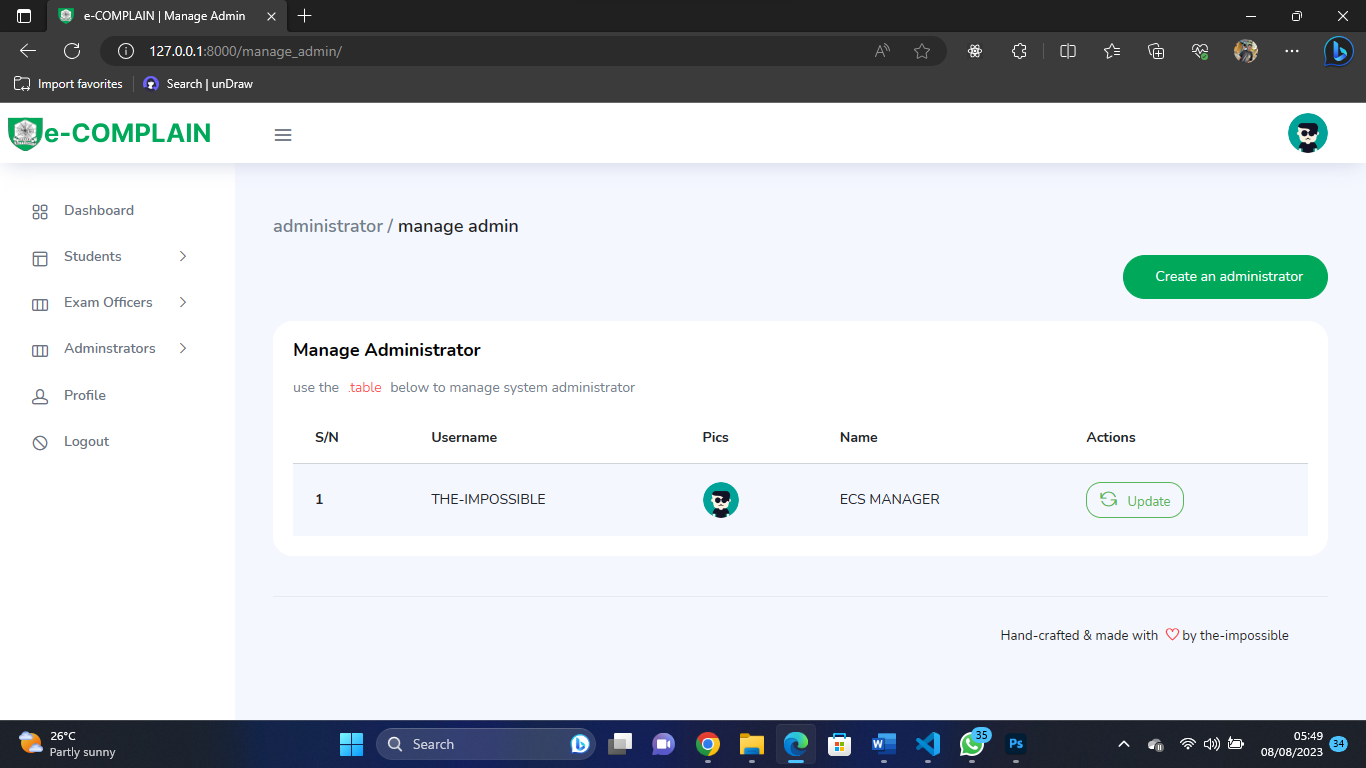


Fig 4.6 Manage Administrative Officers

**Account Profile**

This is the page performs updates (password and information) on the profile of every authenticated user

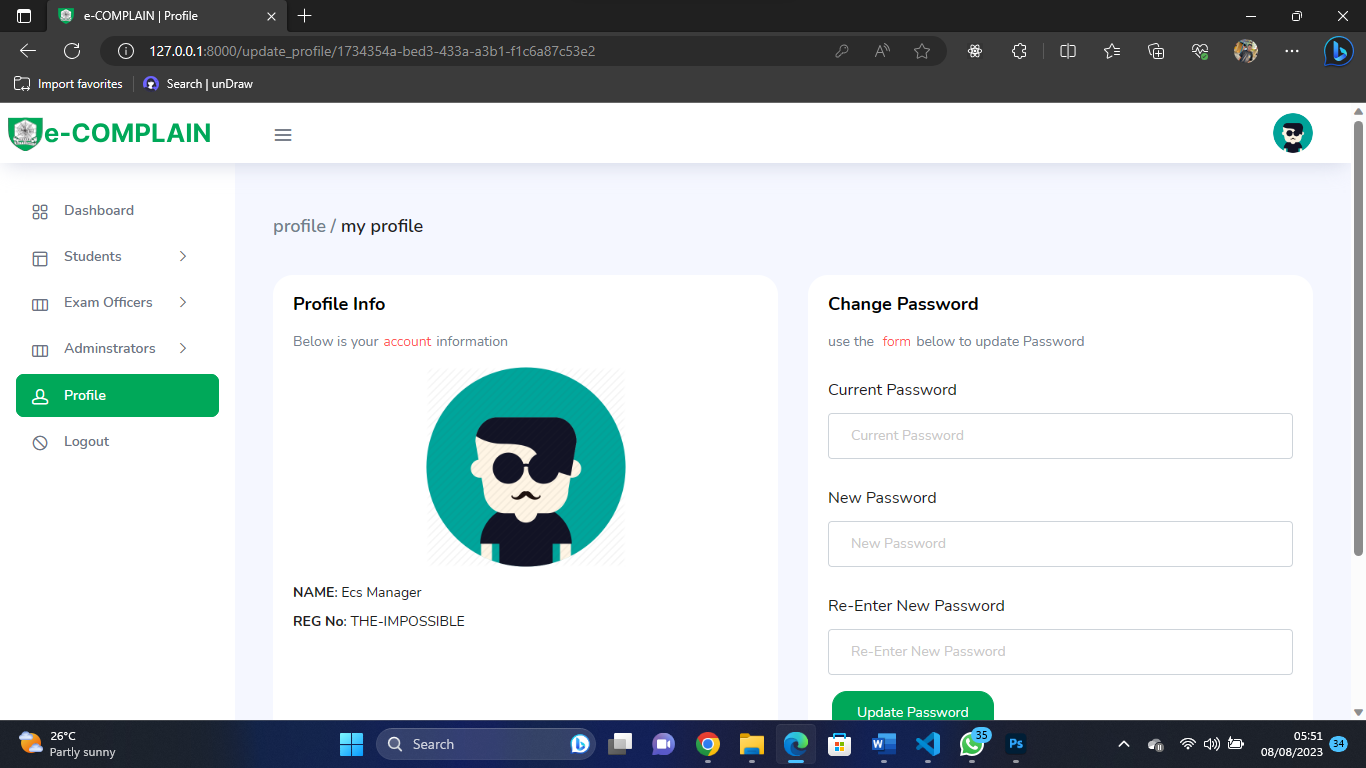


Fig 4.7 Manage Account Profile

**Student Dashboard**

This is the student dashboard, the sidebar shows the available functionality for the student

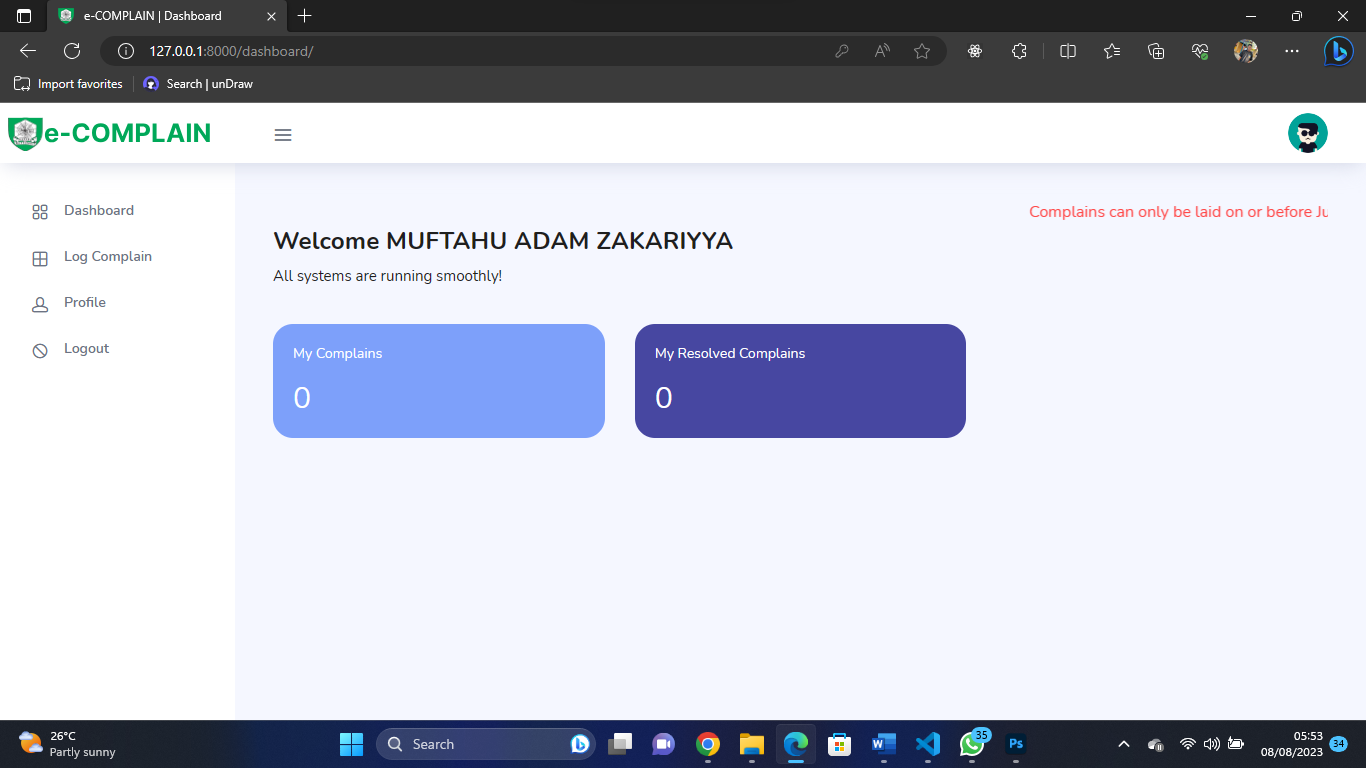


Fig 4.8 Student Dashboard

**Log Complain Page**

The student can use this page to log complaints as well as track the status of his or her complains

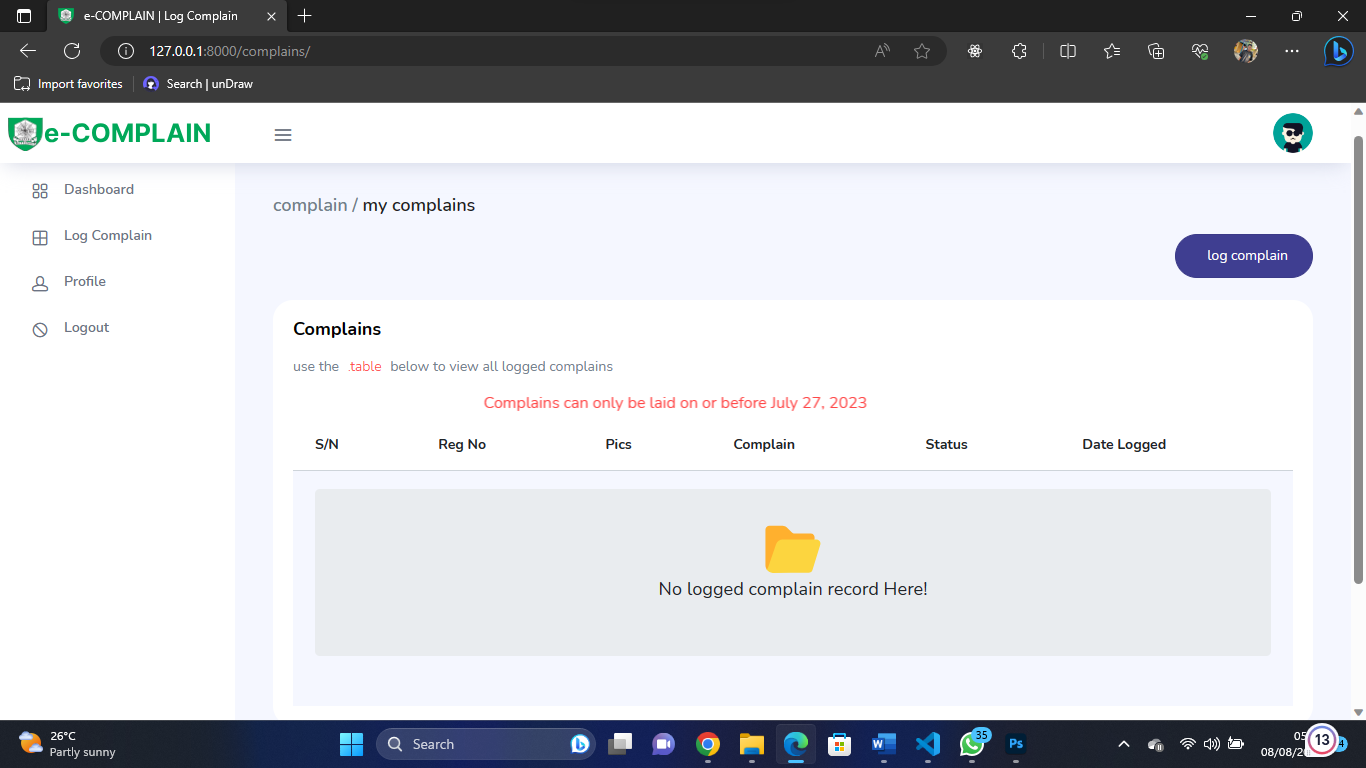


Fig 4.9 Log Complain Page

**Exam Officer Dashboard**

This is an exam officer dashboard, the sidebar shows the available functionality for the user

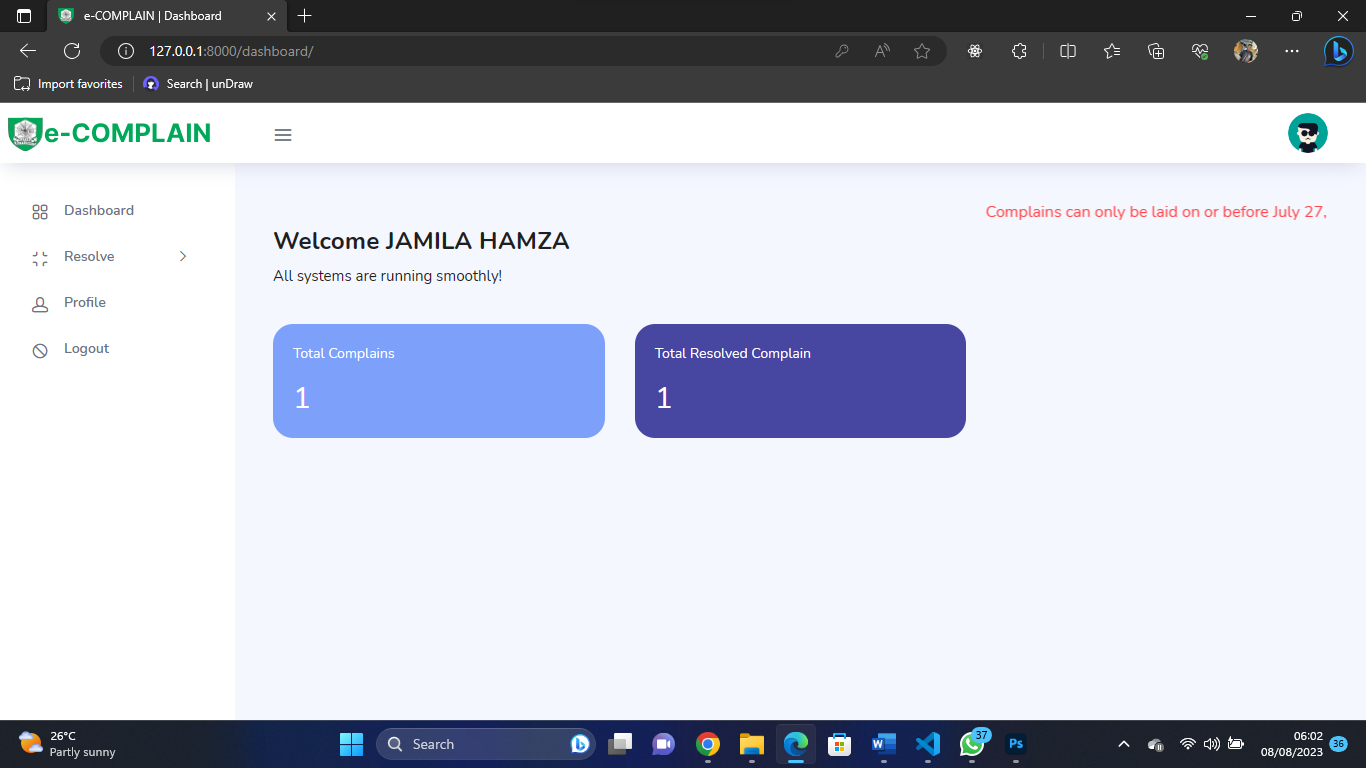


Fig 4.10 Exam Officer Dashboard

**View Complains**

Respective exam officers can utilize this page to view all students complains

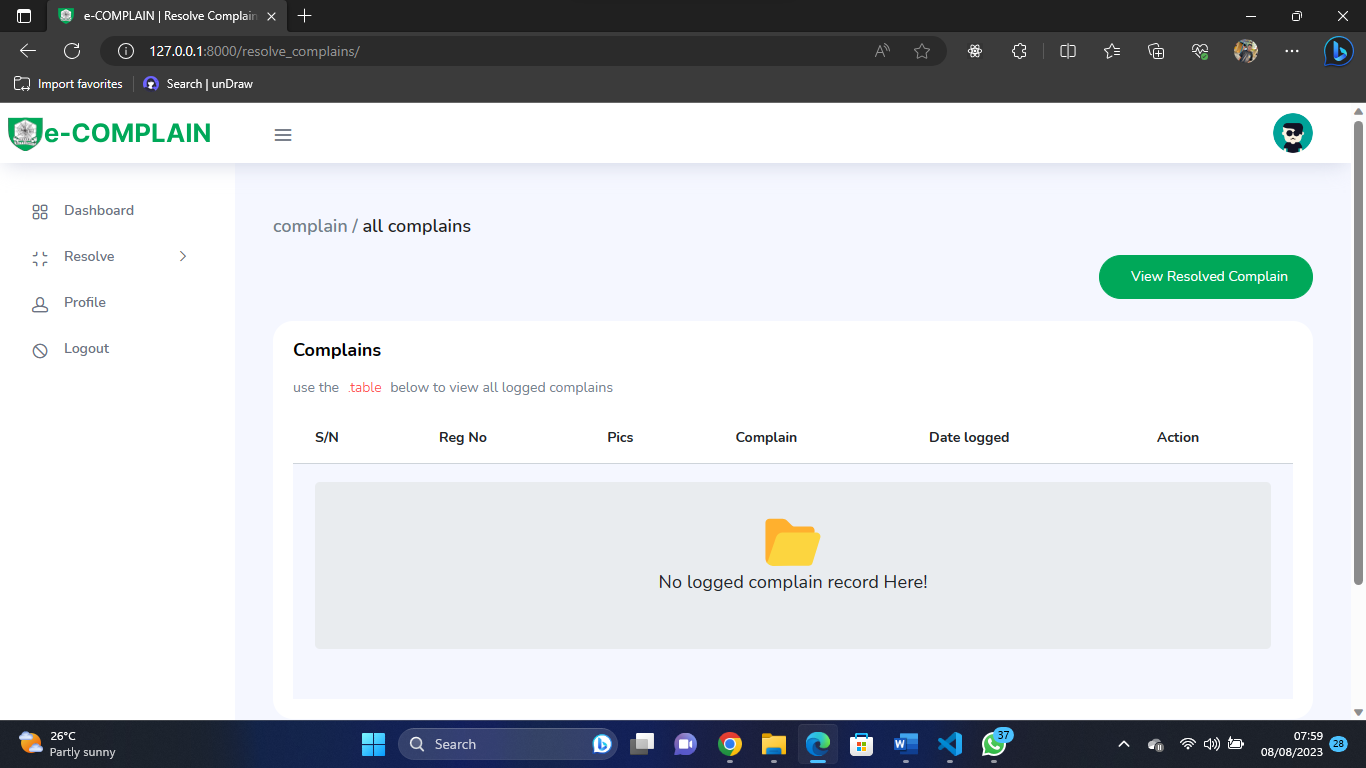


Fig 4.11 View Complains

**Complain Details**

The exam officer can click on a complaint to view the details of the complaint for further rectification

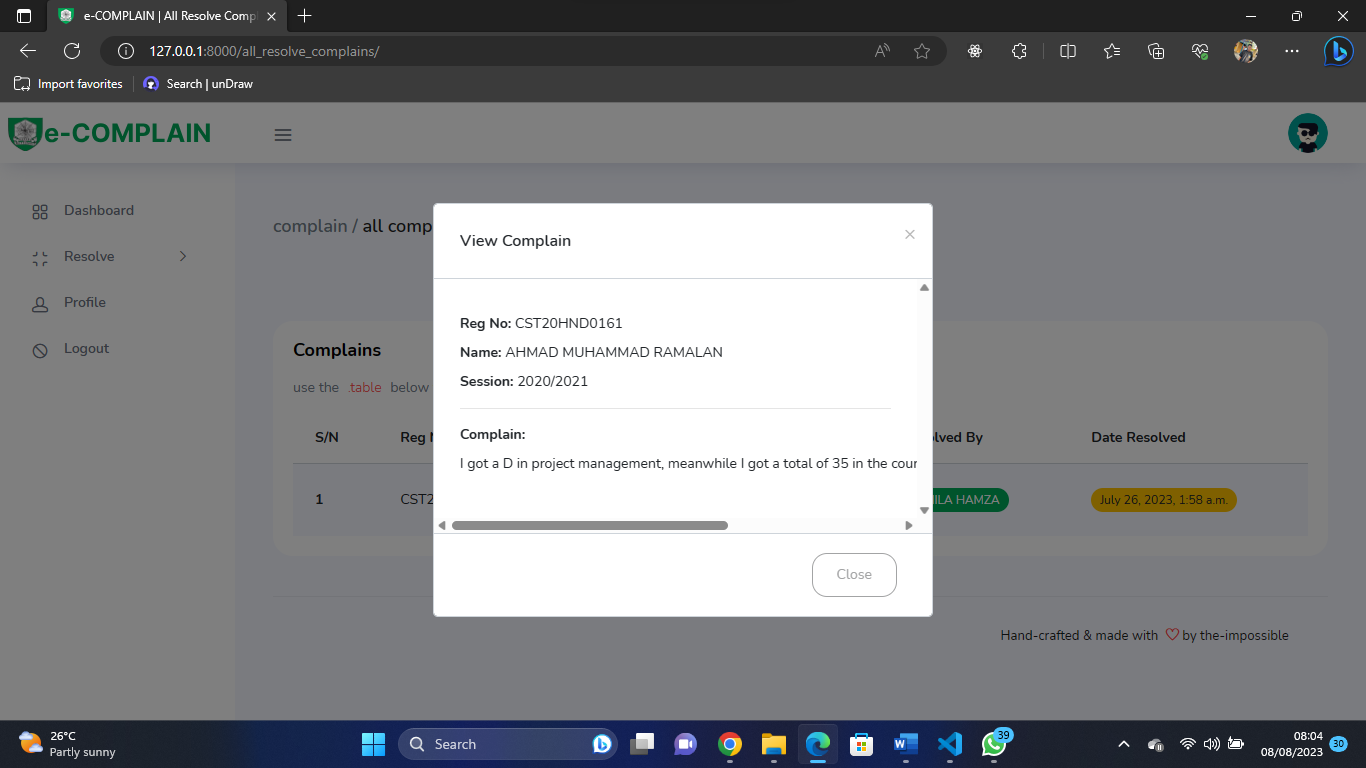


Fig 4.12 Complain Details

**Resolved Complains**

The exam officer can view all resolved complains for future references

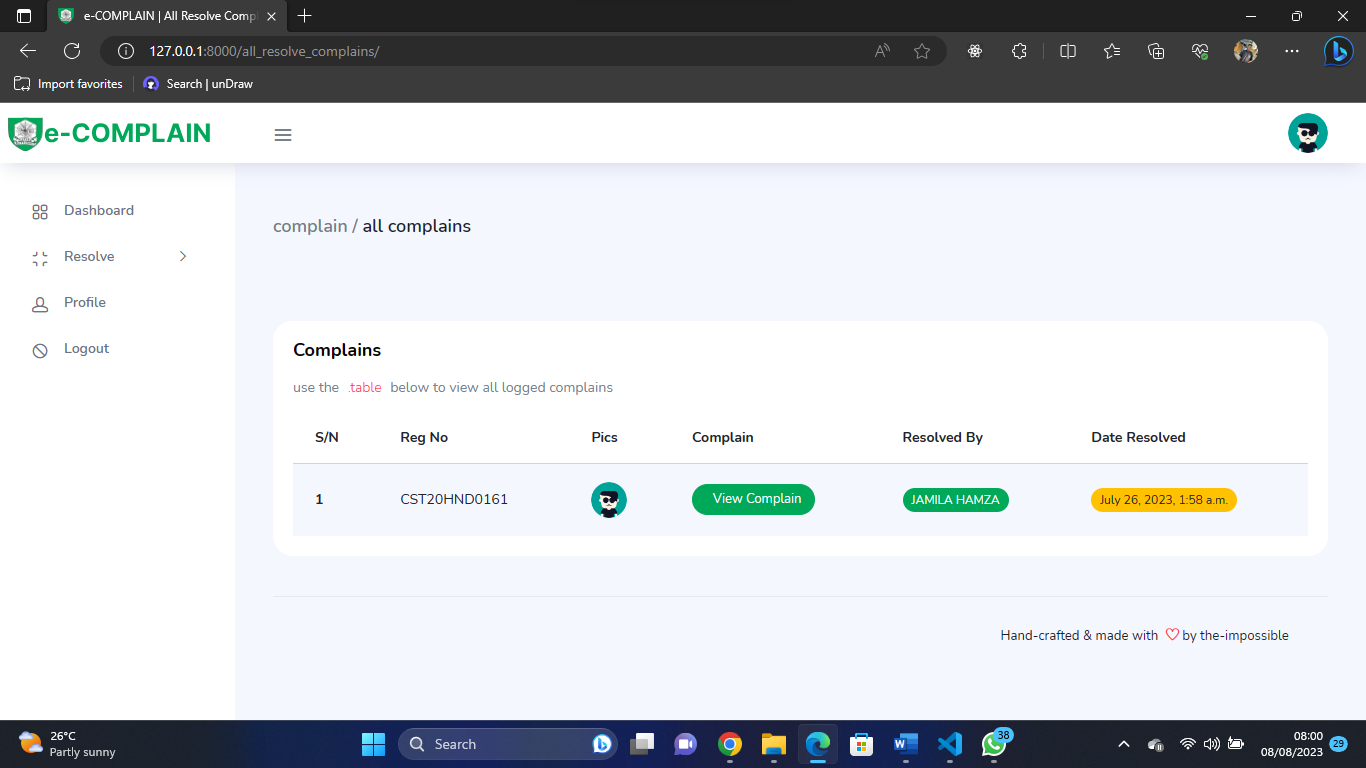


Fig 4.13 Resolved Complains

**CHAPTER FIVE**

**SUMMARY CONCLUSION AND RECOMMENDATION**

**5.1 Summary**

This study delved into the development of a web-based student examination result complaint system for the Department of Computer Science at Kaduna Polytechnic. The need for an efficient and accessible platform to address students' concerns about their examination results was identified. The system aims to streamline the complaint process, providing students with a secure way to submit complaints and track their resolution. Through the study, objectives were set, including the design of a functional platform, testing for effectiveness and efficiency, and evaluating the existing paper-based system. The scope was defined to focus solely on the development of the system, excluding issues related to result validity and authority handling of complaints. The significance of the study lies in providing students with a platform to voice their concerns and enhancing the institution's accountability in addressing these concerns.

**5.2 Conclusion**

In conclusion, the development of a web-based student examination result complaint system represents a significant advancement in enhancing the transparency, efficiency, and accountability of the examination result complaint process. The project aligns with the evolving landscape of educational technology and addresses a critical need for a seamless platform for students to voice their concerns. By streamlining the complaint process and enabling real-time tracking, the proposed system contributes to improved overall satisfaction for both students and the institution. Its successful implementation has the potential to serve as a model for other departments and institutions looking to modernize their complaint-handling mechanisms, thus fostering a more positive and effective educational environment.

**5.2 Recommendation**

Based on the research conducted, it is recommended that the proposed web-based student examination result complaint system be implemented within the Department of Computer Science at Kaduna Polytechnic. This system holds the potential to improve communication between students and authorities, ensuring prompt and accurate handling of examination result-related concerns. Additionally, it is advised that a comprehensive training program be organized for both students and administrative staff to ensure effective usage of the system and to facilitate a smooth transition from the existing paper-based process.

**REFERENCES**

Anusiuba, Ifeanyi & Usman, Karim & Ekwealor, Oluchukwu. (2021). Design and Implementation

of a Tertiary Institution Web-Based Student Complaint Management System. *International Journal of Software & Hardware Research in Engineering. 9*. 10.26821/IJSHRE.9.8.2021.9802.

Kormpho, P., Liawsomboon, P., Phongoen, N., & Pongpaichet, S. (2018). Smart Complaint

Management System. 2018 Seventh ICT International Student Project Conference (ICT-ISPC), 1-6.

Merang, E.K., Ibrahim, A.F., & Jamaluddin, M.N. (2022). Development of a Web-based Water

Supply Complaint Management System. Journal of Computing Research and Innovation.

Raju, R., & Rahman, N. H. A. (2019). Mobile complaint management system (MOCAS).

International Journal of Advanced Trends in Computer Science and Engineering, 8(1.5

Special Issue), 132–135. https://doi.org/10.30534/ijatcse/2019/2681.52019

Singh Bhadouria, L., Kumar Abdul Faisal, N., & Devi, S. (2021). Online Complaint Management

System. *Turkish Journal of Computer and Mathematics Education* (Vol. 12, pp. 5144–5150).

Sudhir, B., Regina, B., & S, S. (2015). Electronic Complaint Management System for Municipal

Corporation. Communications on Applied Electronics, 2(8), 8–11.

<https://doi.org/10.5120/cae2015651817>

teachmint (2022, November 4). *Examination - Complete Meaning & Definition*. Teachmint. <https://www.teachmint.com/glossary/e/examination/>

**APPENDIX**

**Landing page**

{% extends 'base.html' %}

{% load static %}

 {% block title %} HomePage {%endblock %}

  {% block head %} {% include 'partials/head.html' %} {% endblock %}

  {% block body %}

  <body>

    <!-- ===============================================-->

    <!--    Main Content-->

    <!-- ===============================================-->

    <main class="main" id="top">

      {% block nav %} {% include 'partials/nav.html' %}{% endblock %}

      <section class="py-0" id="home">

        <div

          class="bg-holder d-none d-md-block"

          style="

            background-image: url({% static 'assets/img/illustrations/hero-section.png' %});

            background-position: right bottom;

            background-size: contain;

          "

        ></div>

        <!--/.bg-holder-->

        <div

          class="bg-holder d-block d-md-none"

          style="

            background-image: url({% static 'assets/img/illustrations/hero-bg.png' %});

            background-position: right top;

            background-size: contain;

          "

        ></div>

        <!--/.bg-holder-->

        <div class="container">

          <div class="row align-items-center min-vh-md-75">

            <div class="col-md-7 col-lg-6 py-6 text-sm-start text-center mt-5">

              <h1 class="mt-6 mb-sm-4 display-4 fw-semi-bold lh-sm fs-4 fs-lg-6 fs-xxl-7">

                Student Examination Result Complaint System

              </h1>

              <p class="mb-4 ">

                easily and securely submit complains about your examination result

              </p>

            </div>

          </div>

        </div>

      </section>

      <!-- ============================================-->

      <!-- <section> begin ============================-->

      {% comment %} <section>

        <div class="container">

          <div class="row justify-content-center">

            <div class="col-auto mb-5 mb-md-7">

              <h1 class="fw-semi-bold text-warning">

                Our <span class="text-1100">objectives</span>

              </h1>

            </div>

          </div>

          <div class="row">

            <div class="col-sm-6 col-lg-3 mb-4 mb-lg-0 text-center">

              <div class="px-0 px-lg-3">

                <img

                  class="img-fluid mb-4"

                  src="{% static 'assets/img/gallery/researchers.png' %}"

                  width="100"

                  alt="..."

                />

                <h3 class="h5 mb-4 font-base">Researchers</h3>

                <p class="lh-lg">

                  Register online Discover tools and manage alerts Learn about

                  how to access

                </p>

              </div>

            </div>

            <div class="col-sm-6 col-lg-3 mb-4 mb-lg-0 text-center">

              <div class="px-0 px-lg-3">

                <img

                  class="img-fluid mb-4"

                  src="{% static 'assets/img/gallery/librarian.png' %}"

                  width="100"

                  alt="..."

                />

                <h3 class="h5 mb-4 font-base">Librarian</h3>

                <p class="lh-lg">

                  Manage your account View products and solutions Find resources

                  and support

                </p>

              </div>

            </div>

            <div class="col-sm-6 col-lg-3 mb-4 mb-lg-0 text-center">

              <div class="px-0 px-lg-3">

                <img

                  class="img-fluid mb-4"

                  src="{% static 'assets/img/gallery/societies.png' %}"

                  width="100"

                  alt="..."

                />

                <h3 class="h5 mb-4 font-base">Societies</h3>

                <p class="lh-lg">

                  Publish with Wiley Explore our resource library Learn about

                  topics and trends

                </p>

              </div>

            </div>

            <div class="col-sm-6 col-lg-3 mb-4 mb-lg-0 text-center">

              <div class="px-0 px-lg-3">

                <img

                  class="img-fluid mb-4"

                  src="{% static 'assets/img/gallery/authors.png' %}"

                  width="100"

                  alt="..."

                />

                <h3 class="h5 mb-4 font-base">Authors</h3>

                <p class="lh-lg">

                  Submit a paper Track your article Learn about Open Access

                </p>

              </div>

            </div>

          </div>

        </div>

        <!-- end of .container-->

      </section> {% endcomment %}

      <!-- <section> close ============================-->

      <!-- ============================================-->

      <section id="libraries">

        <div

          class="bg-holder"

          style="

            background-image: url({% static 'assets/img/illustrations/libraries-bg.png' %});

            background-position: left bottom;

            background-size: contain;

          "

        ></div>

        <!--/.bg-holder-->

      </section>

  </main>

    {% block script %}

    {% include 'partials/script.html' %}

    {% endblock %}

  </body>

{% endblock %}

**Views.py**

from django.shortcuts import render, redirect, reverse

from django.http import HttpResponseRedirect, Http404, HttpResponse

from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView, TemplateView

from django.contrib import messages

from django.contrib.auth import authenticate, login, logout

from django.contrib.messages.views import SuccessMessageMixin

from django.contrib.auth.mixins import LoginRequiredMixin

from django.utils.decorators import method\_decorator

from django.contrib.auth.hashers import make\_password, check\_password

from django.views import View

from django.utils import timezone

from ECS\_auth.decorators import \*

import csv

import io

import codecs

from django.urls import reverse\_lazy

from ECS\_auth.forms import \*

# Create your views here.

PASSWORD = "12345678"

class HomePage(TemplateView):

    template\_name = "frontend/landing.html"

class DashBoard(LoginRequiredMixin, TemplateView):

    template\_name = "backend/dashboard.html"

    form\_class = EndDateForm

    def get\_context\_data(self, \*\*kwargs):

        context = super().get\_context\_data(\*\*kwargs)

        context["form"] = self.form\_class

        context["date"] = ComplainEndDate.objects.first()

        if self.request.user.is\_superuser:

            context["t\_complains"] = Complains.objects.all().count()

            context["r\_complains"] = Complains.objects.all().count()

            context["t\_students"] = User.objects.filter(is\_office=False, is\_staff=False, is\_superuser=False).count()

        elif self.request.user.is\_office:

            context["t\_complains"] = Complains.objects.all().count()

            context["r\_complains"] = Complains.objects.all().count()

        else:

            context["t\_complains"] = Complains.objects.filter(logged\_by=self.request.user).count()

            context["r\_complains"] = Complains.objects.filter(logged\_by=self.request.user).count()

        return context

    def post(self, request, \*args, \*\*kwargs):

        form = self.form\_class(request.POST)

        if form.is\_valid():

            date = form.cleaned\_data.get('end\_date')

            complain\_date = ComplainEndDate.objects.all()

            if complain\_date:

                complain\_date[0].end\_date = date

                complain\_date[0].save()

            else:

                form.save()

            messages.success(request, "Date Updated successfully!")

        else:

            messages.error(request, form.errors.as\_text())

            return render(request, self.template\_name,

                          context={

                              'form': form,

                          })

        return HttpResponseRedirect(self.get\_success\_url())

    def get\_success\_url(self):

        return reverse("auth:dashboard")

class LoginPageView(View):

    def get(self, request):

        return render(request, 'backend/auth/login.html')

    def post(self, request):

        username = request.POST.get('username').upper().strip()

        password = request.POST.get('password').strip()

        if username and password:

            user = authenticate(request, username=username, password=password)

            if user:

                if user.is\_active:

                    login(request, user)

                    messages.success(request, f"You are now signed in {user}")

                    nxt = request.GET.get('next', None)

                    if nxt is None:

                        return redirect('auth:dashboard')

                    return redirect(self.request.GET.get('next', None))

                else:

                    messages.warning(

                        request, 'Account not active contact the administrator')

            else:

                messages.error(request, 'Invalid login credentials')

        else:

            messages.error(request, 'All fields are required!!')

        return redirect('auth:login')

class LogoutView(LoginRequiredMixin, View):

    def post(self, request):

        logout(request)

        messages.success(

            request, 'You are successfully logged out, to continue login again')

        return redirect('auth:login')

class UpdateOfficerView(LoginRequiredMixin, SuccessMessageMixin, UpdateView):

    model = User

    template\_name = "backend/admin/edit\_delete\_officer.html"

    form\_class = EditExamOfficerForm

    success\_message = 'Updated Successfully!'

    def get\_success\_url(self):

        return reverse("auth:manage\_officer")

class DeleteOfficerView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = User

    success\_message = 'Deleted Successfully!'

    success\_url = reverse\_lazy('auth:manage\_officer')

class ManageAdminView(LoginRequiredMixin, ListView):

    template\_name = "backend/admin/manage\_admin.html"

    form\_class = CreateExamOfficerForm

    def get\_queryset(self):

        return User.objects.filter(is\_staff=True, is\_superuser=True).order\_by('-date\_joined')

    def get\_context\_data(self, \*\*kwargs):

        context = super().get\_context\_data(\*\*kwargs)

        context["form1"] = self.form\_class

        return context

    def post(self, request):

        form1 = self.form\_class(request.POST, request.FILES)

        if 'single' in request.POST:

            if form1.is\_valid():

                instance = form1.save(commit=False)

                instance.is\_staff = True

                instance.is\_superuser = True

                instance.password = make\_password(PASSWORD)

                instance.save()

                messages.success(

                    request, "Administrator created successfully!")

            else:

                messages.error(request, form1.errors.as\_text())

                return render(request, 'backend/admin/manage\_admin.html',

                              context={

                                  'form1': form1,

                                  'object\_list': self.get\_queryset()

                              })

            return HttpResponseRedirect(self.get\_success\_url())

    def get\_success\_url(self):

        return reverse("auth:manage\_admin")

class UpdateAdminView(LoginRequiredMixin, SuccessMessageMixin, UpdateView):

    model = User

    template\_name = "backend/admin/edit\_delete\_admin.html"

    form\_class = EditExamOfficerForm

    success\_message = 'Updated Successfully!'

    def get\_success\_url(self):

        return reverse("auth:manage\_admin")

class DeleteAdminView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = User

    success\_message = 'Deleted Successfully!'

    success\_url = reverse\_lazy('auth:manage\_admin')

class UpdateProfileView(LoginRequiredMixin, SuccessMessageMixin, UpdateView):

    model = User

    template\_name = "backend/profile.html"

    form\_class = EditExamOfficerForm

    success\_message = 'Updated Successfully!'

    def get\_success\_url(self):

        return reverse("auth:manage\_admin")