**CHAPTER ONE**

**INTRODUCTION**

This chapter describes the project and provides some background information. Furthermore, this chapter follows the project's problem statement to clearly understand the project's scope and objectives. This chapter will serve as a blueprint for all subsequent phases of development.

**1.1 Background of The Study**

Before receiving other important documents like certificates and comparable other sensitive things, students in an academic setting are required to show certain critical documents as proof during their clearance (Nneji & Monday, 2018). This means that to gain access, a person must pass through a few departments, each of which is often responsible for carrying out a certain task (Fatima, 2021). This is the situation with students who are in their last year of study and whose results must be cleaned before they can be properly awarded their diplomas.

Since it is customary for final students to walk from one office to another while waiting to be cleared (Usman et al., 2017), and because this practice is usually well documented on paper for record-keeping purposes, even more, every clearing activity is completed manually. However, the majority of organizations understood that the data recorded during student clearance is vulnerable to harm and may even be lost or stolen. In response, several schools began automating the procedure. Due to this, computers are now being used to keep track of the student’s records. It is noteworthy that tertiary institutions have developed an interest in the numerous online educational possibilities due to recent technological advancements (Fatima, 2021).

E-learning is one of the benefits of this significant technological effect on education. E-learning as a concept and as a technology presents a variety of opportunities. Online learning is now more effective and convenient because it has made learning possible from an infinite range and without having to take place in a set structure (class). It has also made people of all ages familiar with and dependent on the internet for information (Nneji & Monday, 2018). By creating online learning platforms, this movement of computerizing education also raises the possibility that an e-clearance system is also necessary (Tunde & Victor, 2021). Since the abilities required to access and understand material online are becoming more widespread and since wireless computing is flexible, it follows that any location can serve as a classroom (Fatima, 2021).

Since the advancement of technology has not only successfully established the possibility of clearance online but has also brought with it the traditional undergraduate and general study program, it is now known that well-organized and competitive integrated online learning resources, as well as those with degree options, have a well-structured online clearance system (Tommy, 2019).

**1.2 Statement of The Problem**

Handling student clearance information and the clearance procedure is done manually. This processing technique takes a long time and is frequently inconvenient. This can be attributed to a variety of issues, including the processing of the clearance form taking longer than anticipated and the absence of key employees who are critical to the clearance's development. The procedure of signing and submitting the essential documents to show that one is a genuine student of the school can be highly hard because there is no predetermined or structured manner of doing so. Students are trapped with the tedious and time-consuming procedure of walking in circles from office to office seeking staff members who can sign, stamp or validate some payment or another. As a result, it is critical to seek a simpler, more effective method of making clearing less difficult and unpleasant. In this regard, the project research provides a solution to the previously described issues by developing an online clearance system for graduating students.

**1.3 Aim of the Study**

The aim of this study is to develop an online clearance system for graduating students at Kaduna polytechnic.

**1.4 Objectives of the Study**

In other to achieve the aim of this project the following objectives are set and considered relevant for its achievement. This includes

1. To review the related literature on the online clearance system for graduating students at Kaduna polytechnic.
2. To design or develop an online clearance system to meet graduate clearance needs.
3. To implement the system and evaluate its efficiency in terms of system information needs / output.

**1.6 Scope of the Study**

This research work would include an investigation of the current methods used in clearance, and an evaluation of the feasibility and potential benefits of implementing an online clearance system. The research work will only be cleared for graduating students and not departmental clearance.

**1.7 Limitations of the Study**

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

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

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

**1.8 Significance of Study**

After completing this project research, the clearance process in Kaduna polytechnic will be computerized, which represents a significant step toward a more efficient and productive clearance process, as the manual means of clearance has been written; it will also make clearance possible from anywhere in the world, as it will be an online system.

**1.9 Project Organization**

This project is made up of five chapters which will be organized in the following order:

Chapter one summarizes the introductory study on the online clearance system for graduating students, including the background of the study, statement of the problem, aim and objectives, the significance of the study, the scope of the study, and limitations of the study. Chapter Two is an overview of related and relevant literature on the topic. Chapter Three describes the research methodology used in this project. That is specific methods that were used in order to achieve the objectives of the system. Chapter Four displays the data analysis and gives concise details of how the system is to be implementedwhile Chapter Five contains the summary and conclusion.

**1.8 Definition of Terms**

1. **Clearance:** This is a status conferred to persons, mainly higher education graduates, military personnel, and public servants, to get access to highly classified items or information**.**
2. **Student:** A person who studies at a higher learning institution**.**
3. **Database:** This saves data over the Internet via a cloud computing provider who is in charge of data administration as a service. It is provided on demand, with just-in-time capacity and pricing.
4. **Online:** This is a term that refers to a person's or device's connection to a network.
5. **Interface:** This is a graphical interface that allows the user to connect with and utilize the application.
6. **Web:** This is the interconnected system of public webpages accessible through the internet.
7. **System:** This is a collection of components that are organized for a common purpose.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

This chapter's goal is to demonstrate how the topic under study relates to previous research, current practice, or other areas of knowledge by citing pertinent works by other scholars that have dealt with a similar issue. In addition, this chapter will provide a synthesis of the existing research on the topic, highlighting areas of agreement, disagreement, and gaps in the literature, to establish the importance of the project topic in the field and to identify areas for further research.

**2.2 Online Clearance System for Graduating Students**

Ramadhan et al. (2020). Design and Implementation of an Open-Source Web-Based Clearance System for Iraqi Educational Institutions. Over the previous few decades, clearance has been done manually. Students who use the manual clearing technique must face a lengthy and laborious process.  Signatures and stamps are used in this process. The process in the Iraqi educational system entails visiting numerous departments, including the library, registrar, accounting, housing, laboratory, storage, sport and artistic activity departments, and the dean's office, only to get the clearance verified. In higher education, final-year understudies who have completed the academic prerequisites to graduate must go through a clearing process before leaving the college.

Furthermore, various web development technologies were used in the implementation of this study report. The website clearly has both a front end and a back end. HTML, CSS, Bootstrap, and JavaScript were the most commonly utilized front-end technologies. The backend, on the other hand, was created through the use of Ajax, PHP scripts, and MySQL for the database. In addition to these technologies, there are certain apps that help with local development on the computer, such as Visual Studio code and XAMP server.

In conclusion, technology makes it easier and faster for students to complete their clearance. Aside from removing paperwork impediments, a web-based system centralizes all data and processes. The system is completely functioning, and each user, namely the Student, Unit Heads, Exam Committee, and System Administrator, may carry out their responsibilities with ease. The system offers numerous advantages over the old approach, such as time savings, cost savings, and data loss.

Grace et al. (2018). An Improved e-Clearance Management System for Graduating Students in a University Environment. This present approach is entirely paper-based and requires personal presence. It entails the graduate student completing the clearance form and then visiting numerous clearing sections such as the financial office, library, laboratory, and department. This technique requires students to be physically present in all clearing parts. Other drawbacks of the manual approach are the amount of paperwork required, the delay in processing clearance papers, and there is the considerable danger that information will fall into the hands of an unauthorized or incorrect person who will modify the information. Due to the absence of key personnel in some clearing units, students may be required to return to the clearance unit before moving to other units. As a result of this system, some students' files are lost, and so on.

Moreso, this study seeks to improve the present system by investigating its flaws. Front-End Languages utilized include HTML for generating web pages, CSS for designing web pages, JavaScript for programming web pages, Sublime Text 3 for code editing, PHP for the back end, MySQL for the database, and the XAMPP servers for system design and implementation. Structured System Analysis and Design is the approach employed in this work (SSADM).

In conclusion, an enhanced electronic clearance management system is proposed in this research. The system will provide graduates with a trouble-free clearing experience. By replacing the manual and semi-manual techniques with the suggested system provided in this study, the cost and time used during the clearing process would be reduced. It has been discovered that if this better method is applied, it would boost efficiency and dependability for both students and the institution.

Jonathan et al. (2019). Development of an Online Clearance System for an Educational Institution. Many institutions do clearance mostly at the point of final exit or collection of final findings. It is unfortunate, however, that most institutions continue to carry out this critical function manually. There are several concerns about manual clearance, including the clearing officer's unavailability when needed, the number of offices and departments to visit, which requires a lot of effort moving around, the process is time and effort-consuming, the clearing officer may cleverly delay expecting some kind of gratification from the students, the queue for clearance may be quite long, students becoming tired while waiting, and the inability to check system for student records due to power failure or faulty system and so on.

Furthermore, the system was built with HTML (Hypertext Markup Language), PHP (Hypertext Preprocessor), and JQuery. They make web-based application development easier. The MySQL server was used to maintain the databases.

Finally, this work improves on previous systems in several ways, including improved performance, user-friendliness, and ease of use. Overall, the program addresses the majority of the flaws indicated by the manual approach. The application was thoroughly tested using industry-standard testing procedures. This was done both throughout the development of the system's many components and after the final application was completed. Local mistakes discovered during unit testing were fixed, and interface concerns were minimized during integration testing.

Mohammed (2021). Online clearance management system. The existing clearance processing system is a manual method that requires a lot of human effort, resources, time, and security. Here, the student will visit all clearing agencies with a form that they will fill out and get signed by the respective offices. The fact that these documents have been signed demonstrates that the users have been vetted. This procedure takes several days to complete and causes great stress for both users and staff who supply the clearing system. In general, the aforementioned difficulty rendered the entire Clearance procedure time-consuming and unpleasant. As a result, an automated clearance system is necessary to remove the bottleneck created by the current human process.

Furthermore, the designed method will serve as a more dependable and effective way of completing student clearance, eliminating all types of time delay and tension, and allowing you to grasp the procedure involved, as well as how to conduct your clearing online. This project work uses data collected from the institution, materials and publications from many authors, and software designed to efficiently meet the project's goal. This project's implementation is carried out using PHP and MYSQL as the database. Finally, it will achieve all of the objectives and will be recommended for all departments at Haramaya University and other postsecondary institutions.

Agbo-Ajala and Makinde. (2017). A Web-Based Database-Driven Students' Clearance System. The clearance procedure encompasses the Academic Department, Faculty, Bursary, Students Affairs, Library, Hostel, Sports Department, Health Center, and Registry (Exams and Records). A student may get his or her graduation certificate only when he or she has been cleared. The existing system is plagued by the following issue: Inadequate database management, Inadequate user interface, and Information processing are slow. Information is vulnerable to tampering and moisture. Distinguished by the employment of a pen and paper, It increases administrative personnel stress and workload.

Moreso, the Student Clearance System was created with PHP and MySQL and implemented with the data obtained. It was able to handle student clearance processes across all departments and units, and it solved the shortcomings of the manual approach, which primarily lacked a centralized repository for the clearance process. Others complain that the procedure is sluggish, awkward, and frustrating. As a result, the clearance application may be used to record and confirm whether or not a student is cleared to disconnect. Clearance applications are tracked, and the status can be verified at any moment.

**2.3 Summary of Related Literature Reviews**

|  |  |  |
| --- | --- | --- |
| **Author & Year** | **Title & Description** | **Merit and Demerits** |
| Ramadhan et al. (2020). | Design and Implementation of an Open-Source Web-Based Clearance System for Iraqi Educational Institutions.  The system  enables students to do their clearance much easier and faster. | The whole data  and the process is centralized with a web-based system  Some areas of the system lack maintenance. |
| Grace et al. (2018). | An Improved e-Clearance Management System for Graduating Students in a University Environment.  The e-clearance system is a web-based method that enables graduating students to complete their final year clearance electronically or online. | The system will provide graduates with a smooth clearance experience.  The system may lack scalability probably due to the influx of users. |
| Jonathan et al. (2019). | Development of an Online Clearance System for an Educational Institution.  The project uses a case study method to develop a functioning prototype of a comprehensive manual system for a prominent institution of learning in Southwest Nigeria. | The new approach will minimize the amount of time and effort spent on student clearance, as well as the expense incurred by the institution on paper.  The system is strictly a web-based application. |
| Mohammed (2021). | Online clearance management system.  The system will replace the manual method of clearance during withdrawal, summer vacation, and graduating time for students. | The system eliminates all sorts of time delays and tension while also allowing you to comprehend the procedure.  Poor user interface. |
| Agbo-Ajala and Makinde. (2017). | A Web-Based Database-Driven Students' Clearance System.  The project's goal is to design and build an online student-clearing system utilizing PHP and MySQL that will reduce human delays and provide a central repository for students to be cleared. | The system creates a central repository for students that need to be cleared.  The usability of the system is poor. |

**2.4 Analysis of the Current System**

This present approach is entirely paper-based and requires personal presence. It entails the graduate student completing the clearance form and then visiting numerous clearing sections such as the financial office, library, laboratory, and department, among others. This technique requires students to be physically present in all clearing parts.

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

The manual clearance system currently in use is time-consuming, costly, and has additional drawbacks, such as requiring students to drive a considerable distance to the school to receive clearance. The process frequently deviates from the original goal and has, in most cases, cost graduates their jobs or attracted additional fees

**2.5 Analysis of the Proposed System**

The proposed system is intended to address the shortcomings of the present system. The upgraded technology is highly useful for replacing human operators in activities that require demanding physical or repetitious labour. The system automatically creates, reads, updates, and deletes redundant data.

**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 a clearance system for graduating students at Kaduna polytechnic.

**3.2 Methods of Data Collection**

Before developing any system, collecting data and facts about the existing system is critical to understand what is going on. This research was carried out using three methods.

1. Observation of the Work Environment
2. Interview
3. Documentation

**3.2.1 Observation of the Work Environment**

By observing how the manual system worked, this strategy was used to collect information and data for this study. A thorough examination revealed the most obvious flaws in the current system. When using the observational technique, the environment in which the observation is made can be changed in a variety of ways.

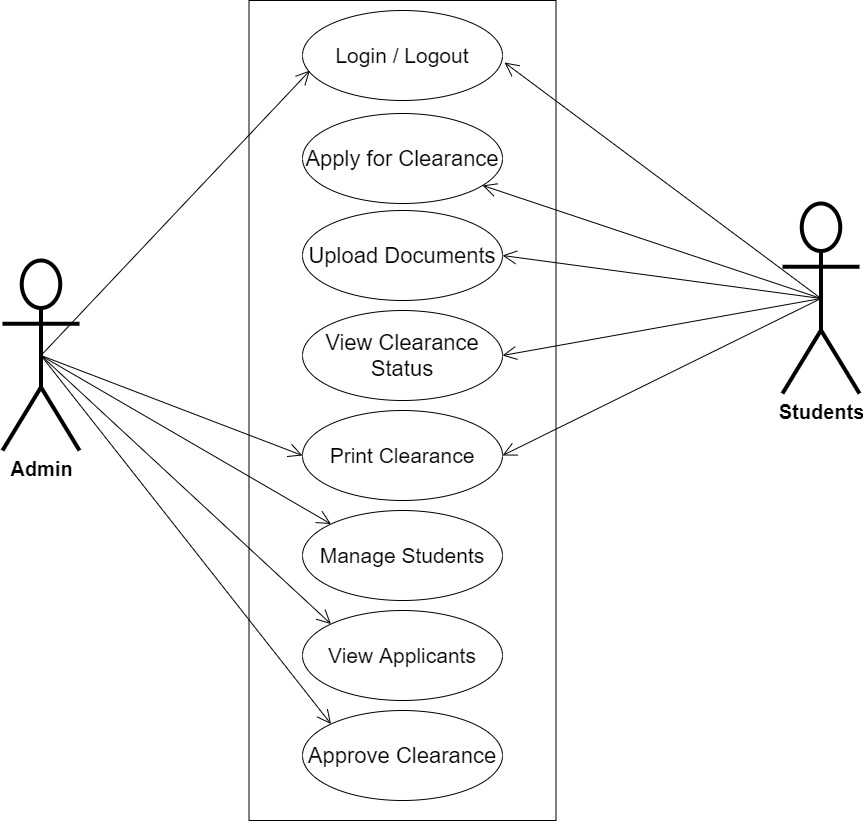
**3.2.2 Documentation**

Secondary data gathering includes documentation. Journals, manuals, previous work, publications, and other sources are used in this manner. This data-gathering strategy is chosen because it allows for comparison with previous research. This includes the internet, which is a tool for data collecting. The internet was utilized to research complex or unclear problems.

**3.3 System Modeling**

A system model is a conceptual model of a system that explains and represents it. A system is any interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the creation of this contemporary system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

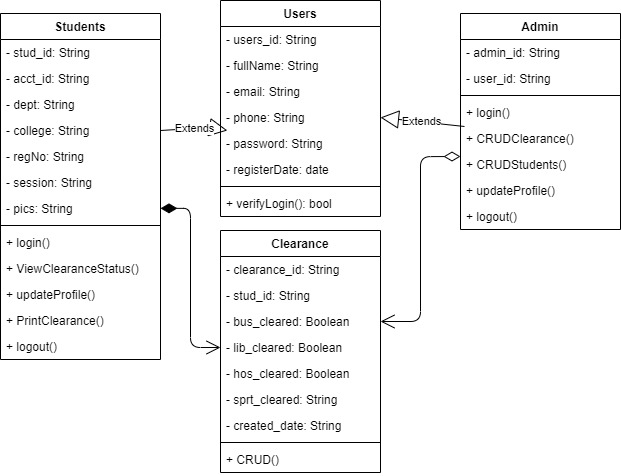
**3.3.1 Use Case Diagrams**

Use cases are collections of interactions between the system and the user. Use case diagrams are used to graphically depict a system's functionality in terms of its actors, goals (represented as use cases), and dependencies among those use cases.

**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagrams**

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



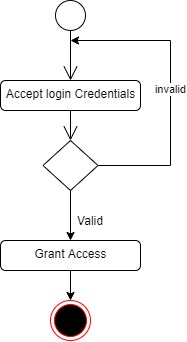
**Fig 3.2 System Class Diagram**

**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**

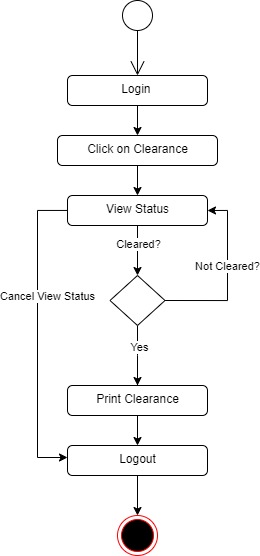
The process for gaining access to the system is depicted in the diagram below; the email address and password must be accurate to gain access.



**Fig 3.3 Login Activity Diagram**

**Clearance Status**

The process for viewing clearance status by a student is depicted below, to view the student has to be authenticated.



**Fig 3.4 Clearance Status Activity Diagram**

**3.4 Database Design**

The logical explanation of how data is kept in the computer's memory is called input specification. The freedom experienced in using the system, as well as the convenience of retrieving and reading the data and assuring applicability across the internet, make SQL standards essential for ensuring that structured data is uniform and independent of applications. Some of the input specifications employed in this project work are presented below.

1. Users Table: contains basic information about all system users.
2. Clearance Table: contains every system-saved student clearance 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 invigilators |
| email | Varchar | No |  | 100 | User email address |
| password | Varchar | No |  | 128 | User Password |
| fullName | Varchar | No |  | 60 | User full name |
| phone | Varchar | No |  | 11 | User phone |
| Register\_date | DateTime | No |  | 20 | User registration date |

**Table 3.2 Clearance** **Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| clearance\_id | Varchar | No | PK | 32 | Unique string for identifying clearance |
| stud\_id | Varchar | No |  | 100 | A unique reference to the student table |
| bus\_clearance | Boolean | No |  | 100 | Bursary clearance status |
| lib\_clearance | Boolean | No |  | 128 | Library clearance status |
| hos\_clearance | Boolean | No |  | 10 | Hostel clearance status |
| sprt\_clearance | Boolean | No |  | 14 | Sports clearance status |
| created\_date | DateTime | No |  | 20 | Date the clearance was initiated |

**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**

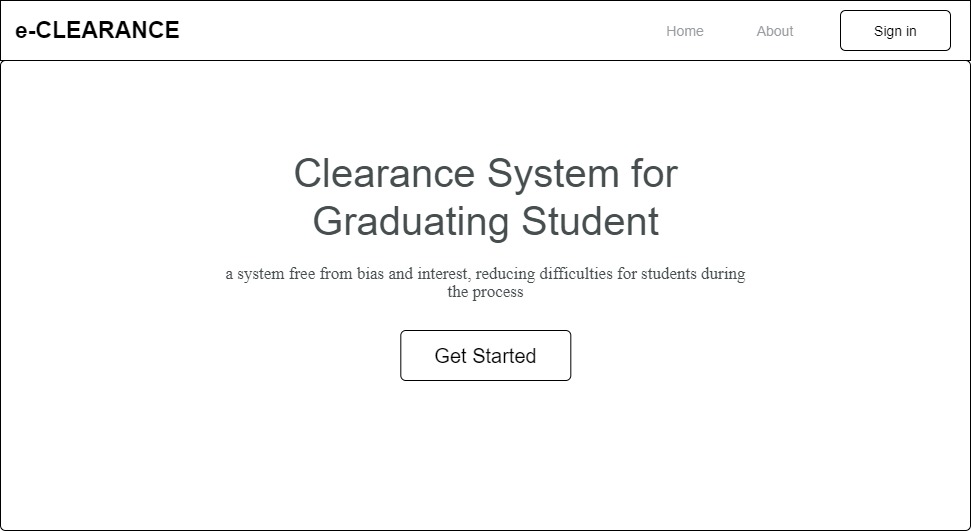
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User\_id** | **Email** | **Password** | **FullName** | **Phone** | **registerDate** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.4 Clearance** **output design table**

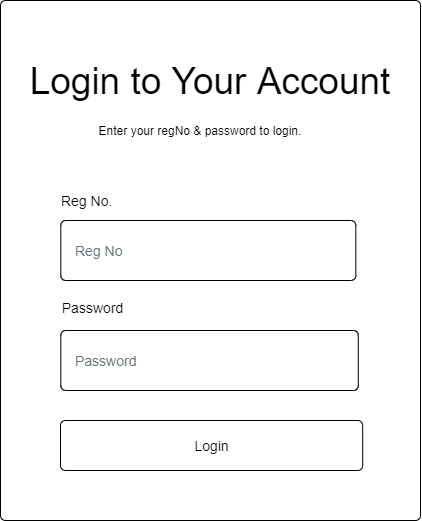
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Clearance\_id** | **Stud\_id** | **Bus\_clearance** | **Lib\_clearance** | **Hos\_clearance** | **Sprt\_clearance** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**3.6 Input & User Interface Design**

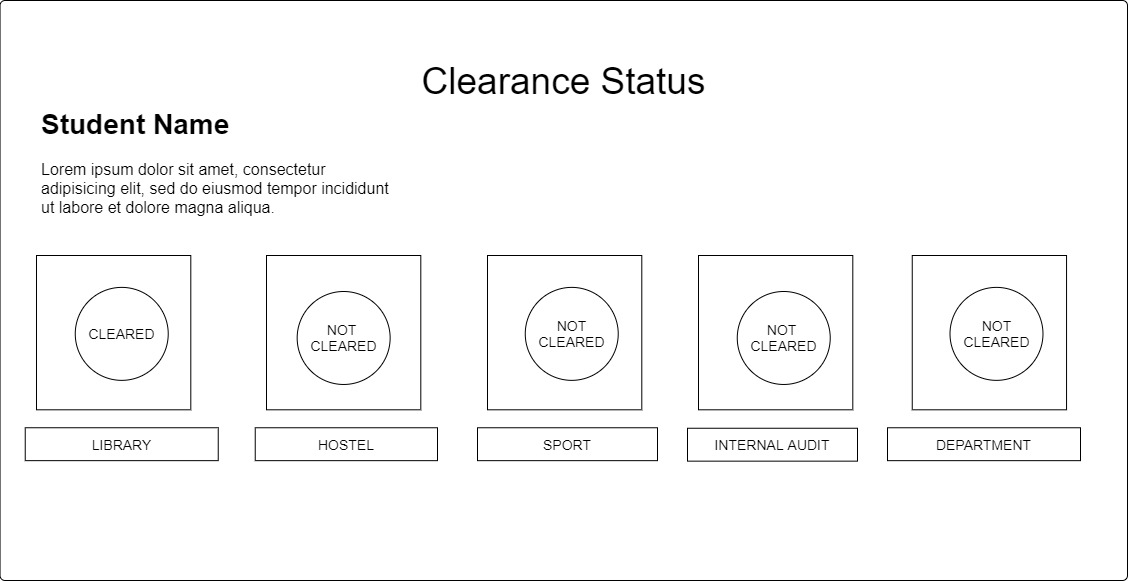
This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be fully secured, thus authentication will be required to see various levels of the information. To help with the designs, a mid-fidelity wireframing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



**Fig 3.5 Home Page**



**Fig 3.6 User Login Form**



**Fig 3.7 Clearance Status Page**

**3.7 System Requirement**

Every piece of software that is 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 8 GB of RAM (Random Access Memory) installed.
2. Minimum of intel core i3 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. Python Installation.
3. Vs. Code installation.
4. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This research work will be a web-based application and will be implemented on a relational database system (SQLite). HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript will be employed in the front end while Django(python) will be employed for the backend programming. The above are the modern languages used in implementing this system.

**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**

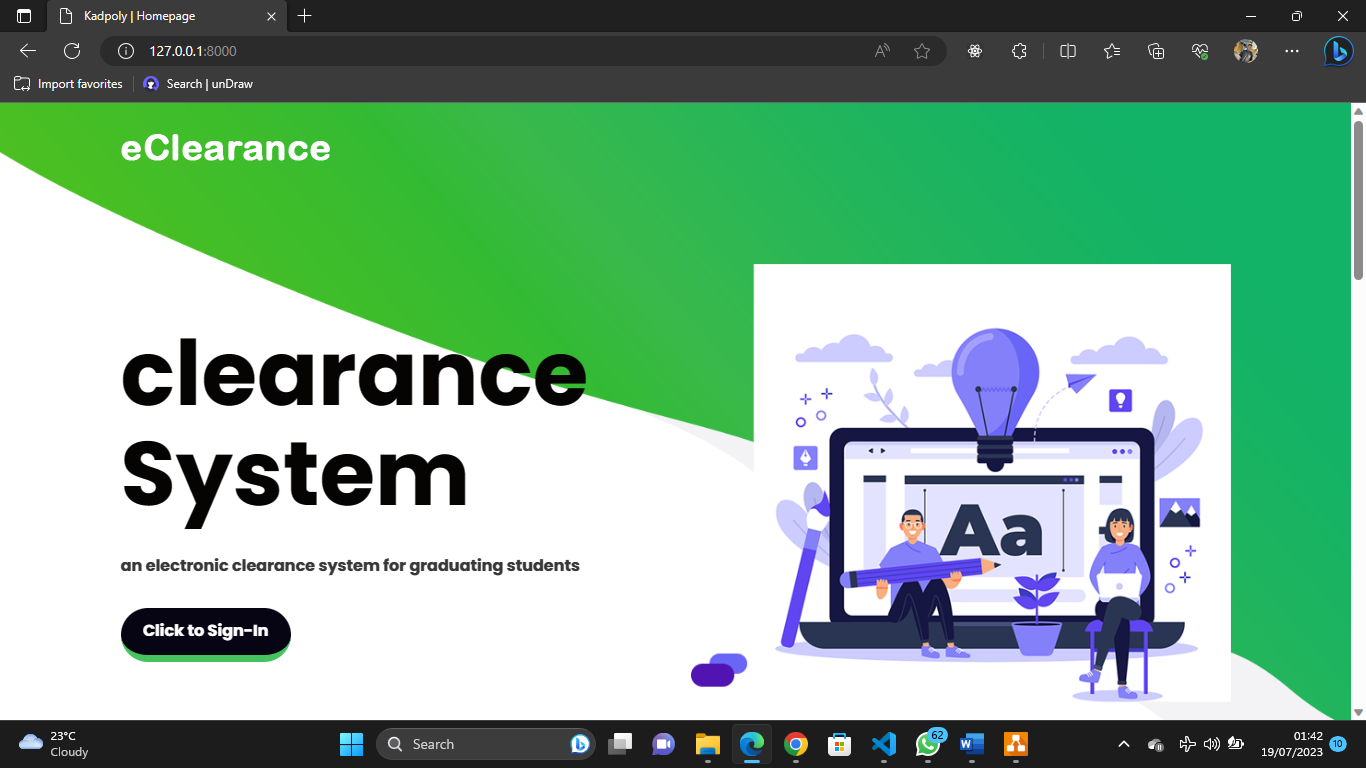
The page serves as a gateway to navigate and explore the various sections of the website

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

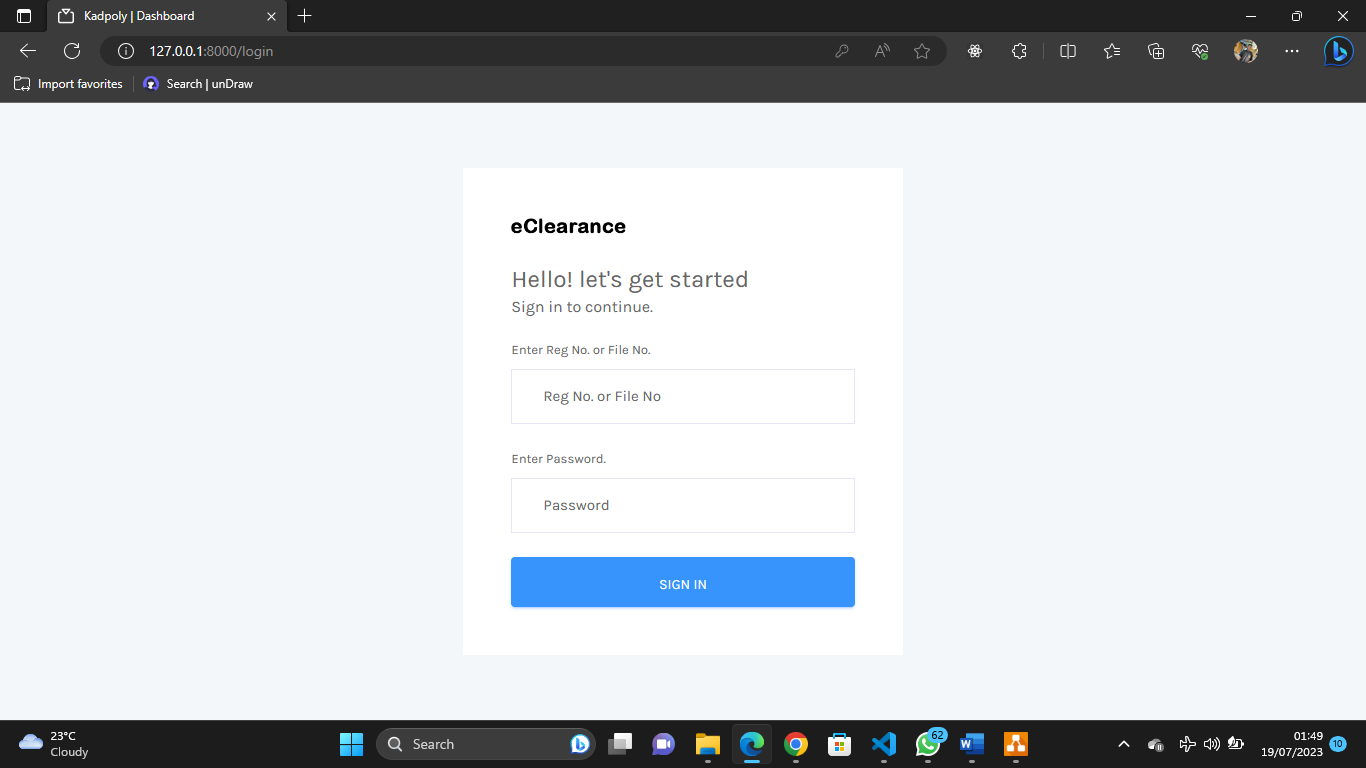


Fig 4.2 User Login

**Admin Dashboard**

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

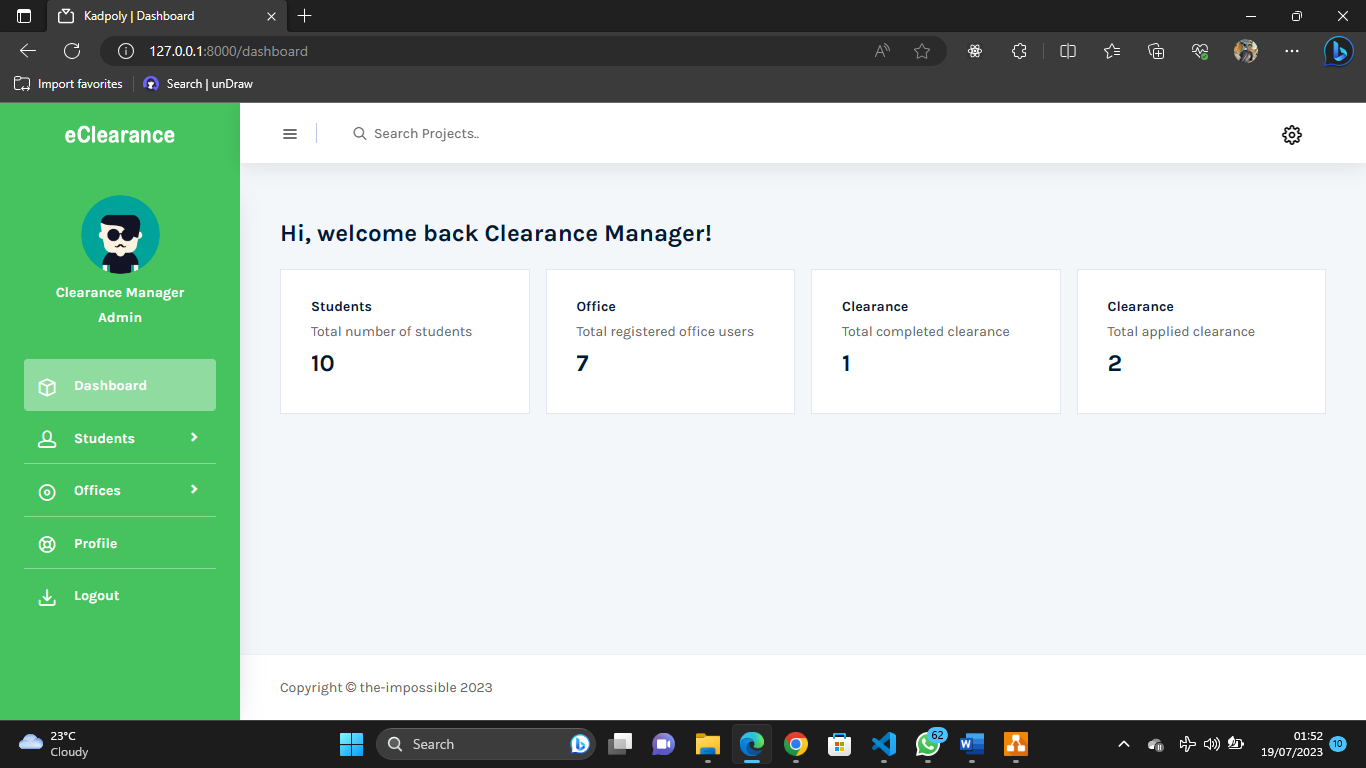


Fig 4.3 Admin Dashboard

**Manage Student’s Account**

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

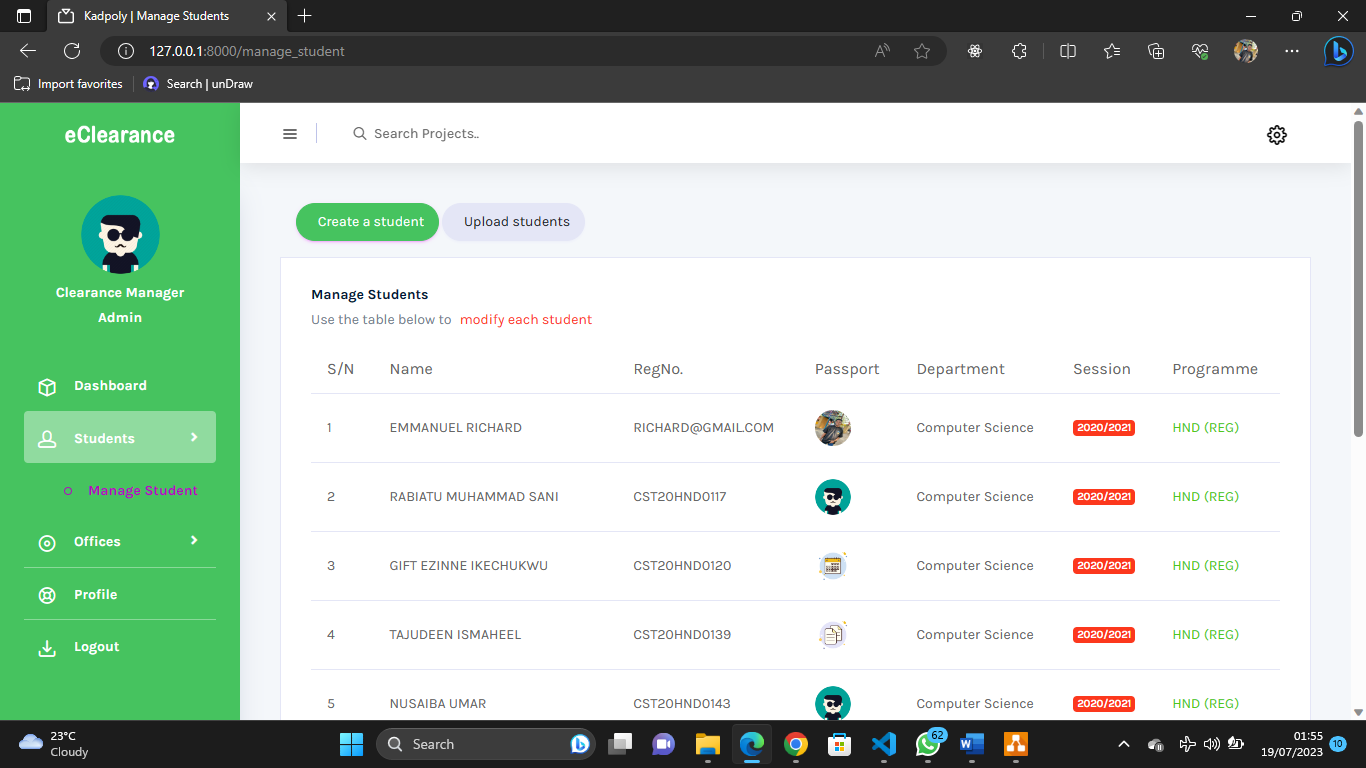


Fig 4.4 Manage Student’s Account

**Manage Administrative Offices**

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

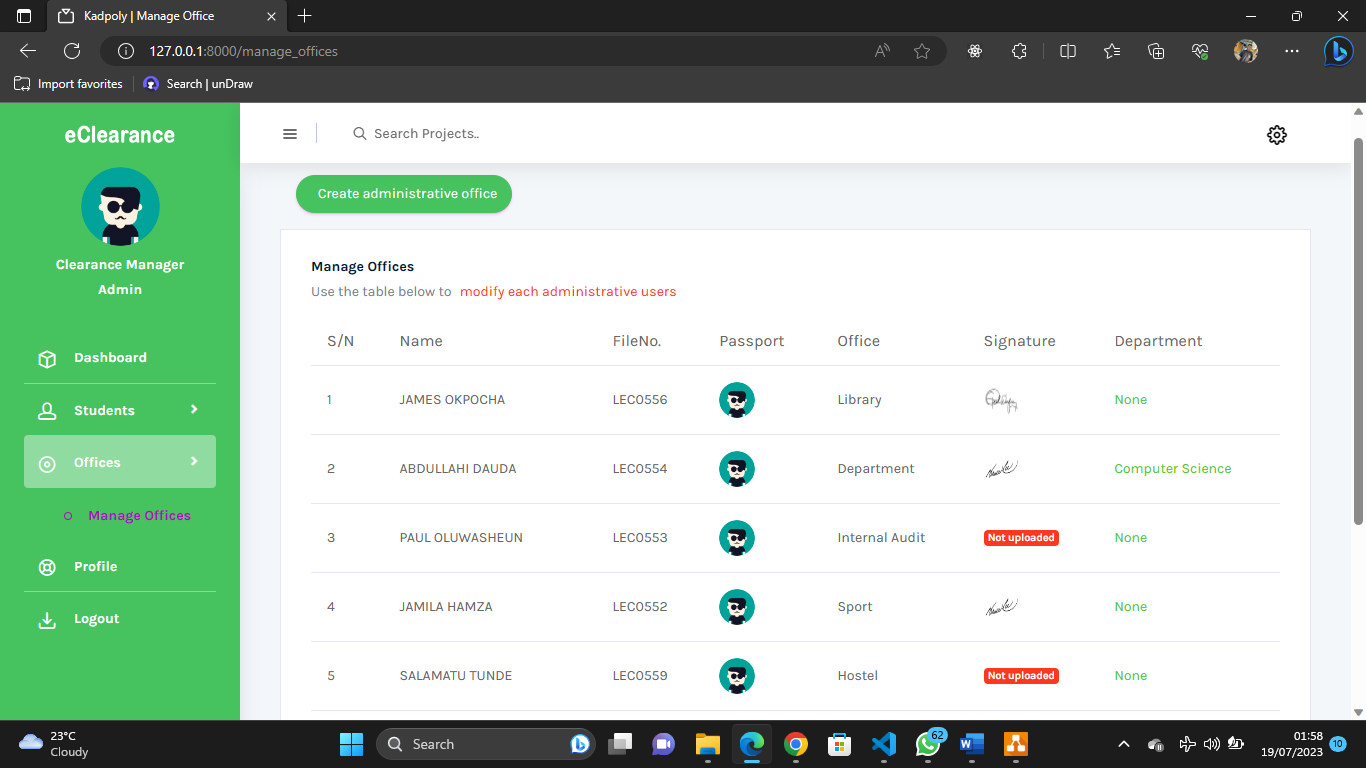


Fig 4.5 Manage Administrative Offices Accounts

**Account Profile**

This is the page performs updates on the profile of every authenticated user

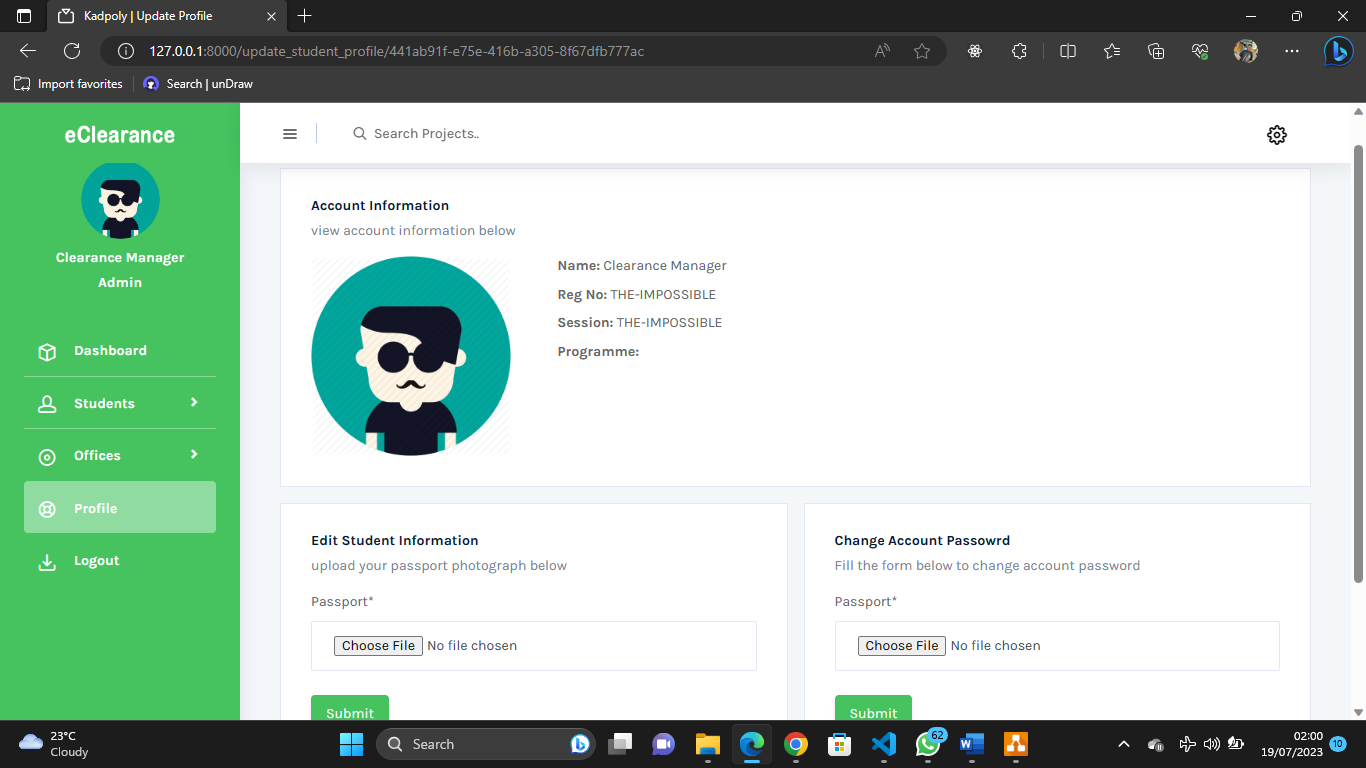


Fig 4.6 Manage Account Profile

**Student Dashboard**

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

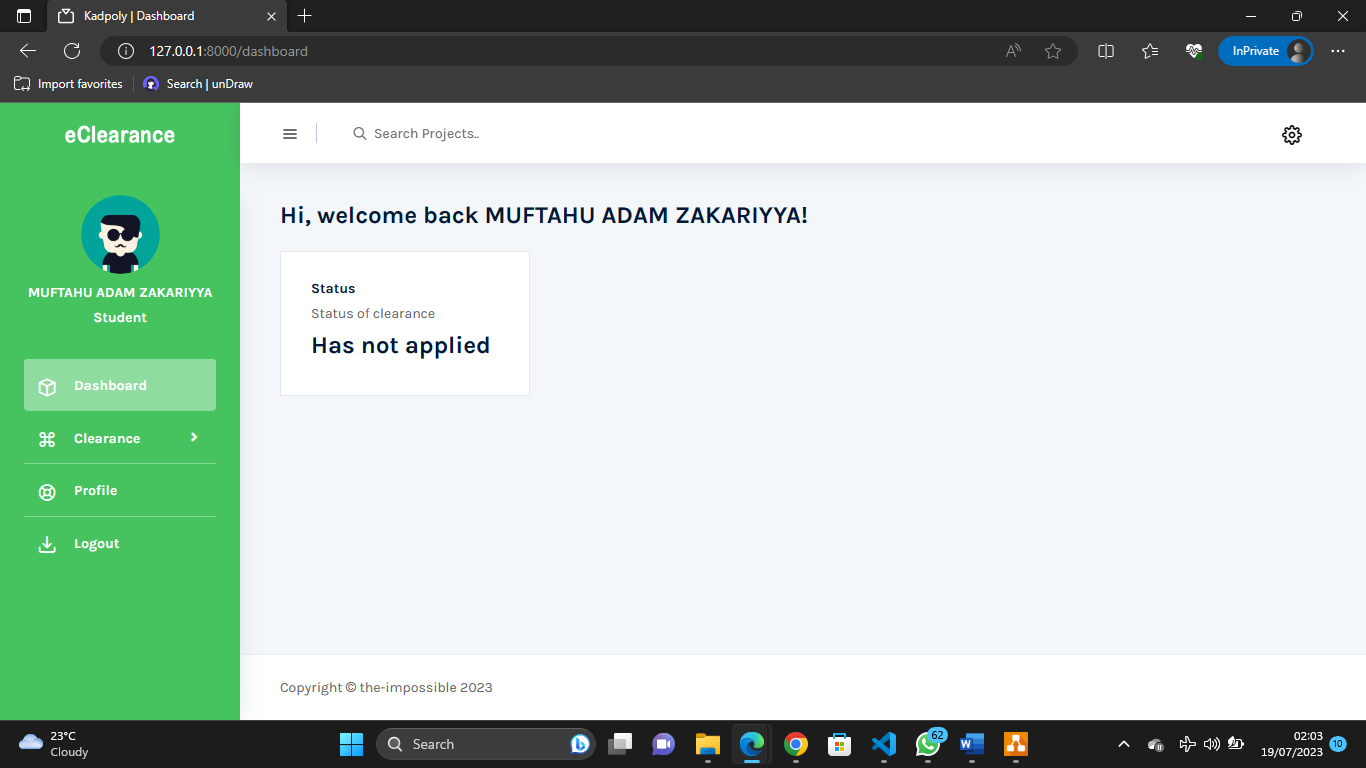


Fig 4.7 Student Dashboard

**Student Clearance Status**

The student can use this page to track his or her clearance status

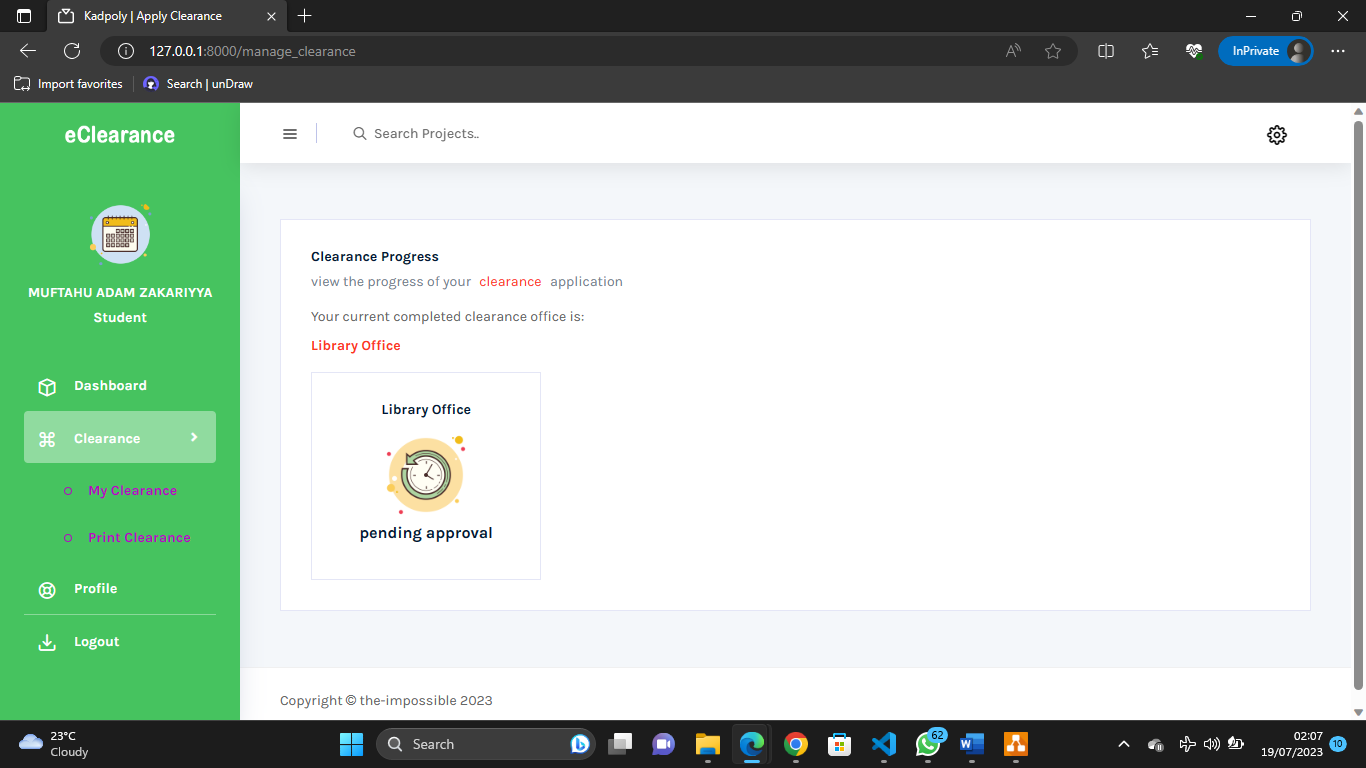


Fig 4.8 Student Clearance Status

**Administrative Office Dashboard**

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

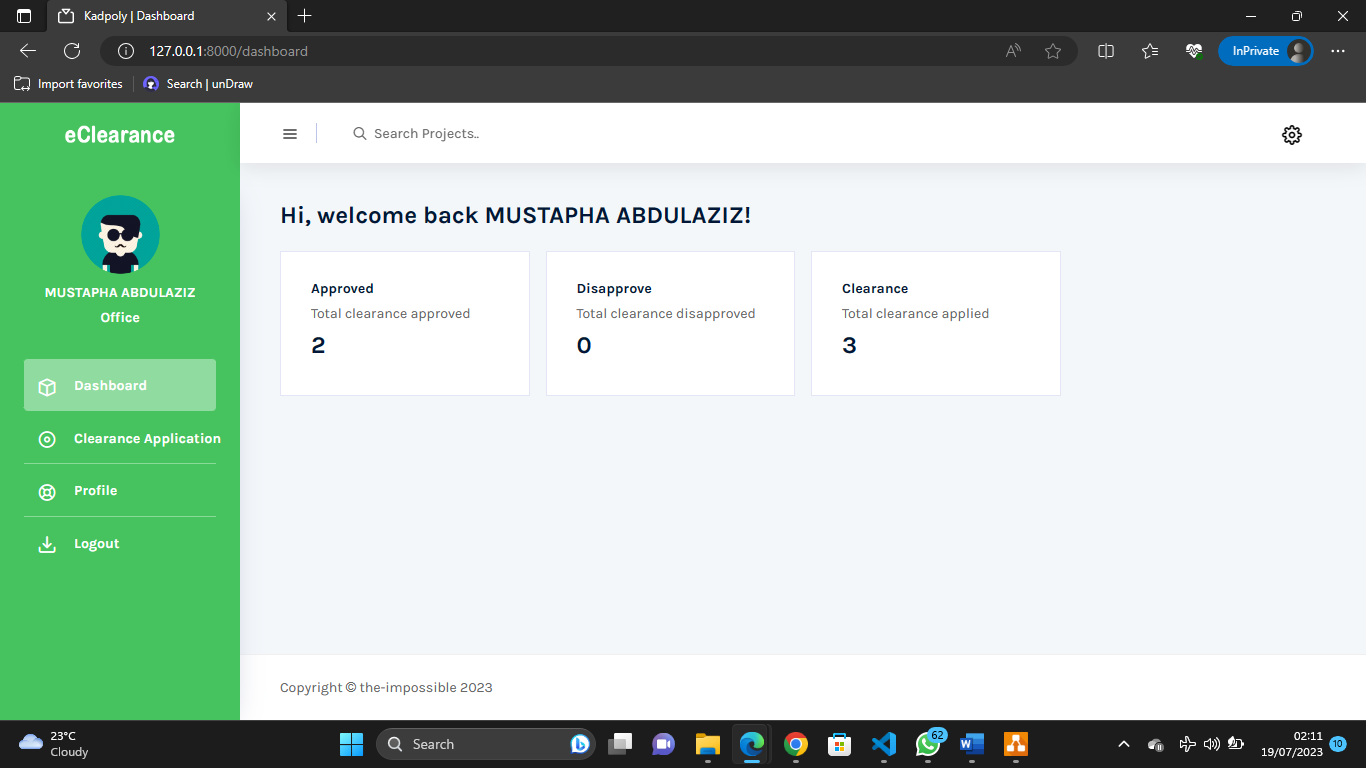


Fig 4.9 Administrative Office Dashboard

**Clear Student**

Respective administrative offices can clear students if the student meets the requirements

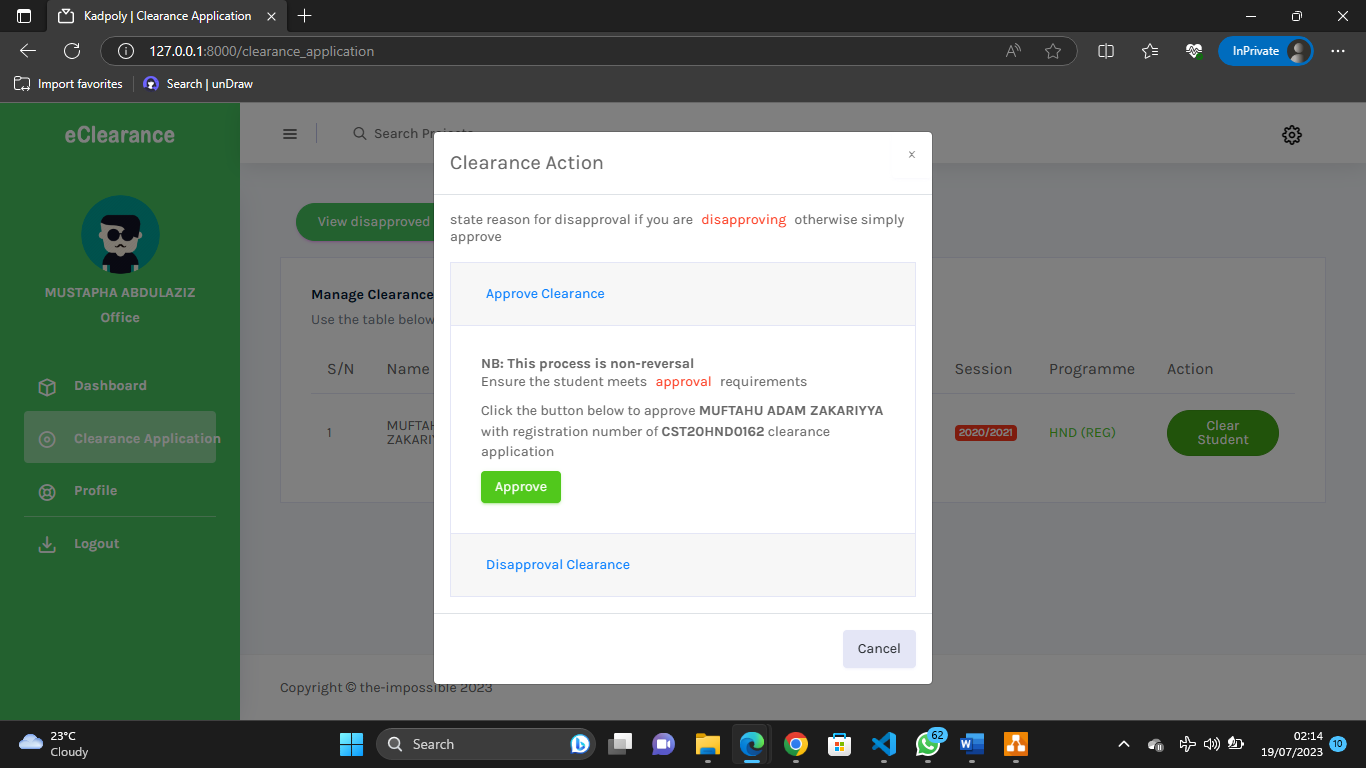


Fig 4.10 Clear Student

**View Files**

Administrative office can view student clearance attached files

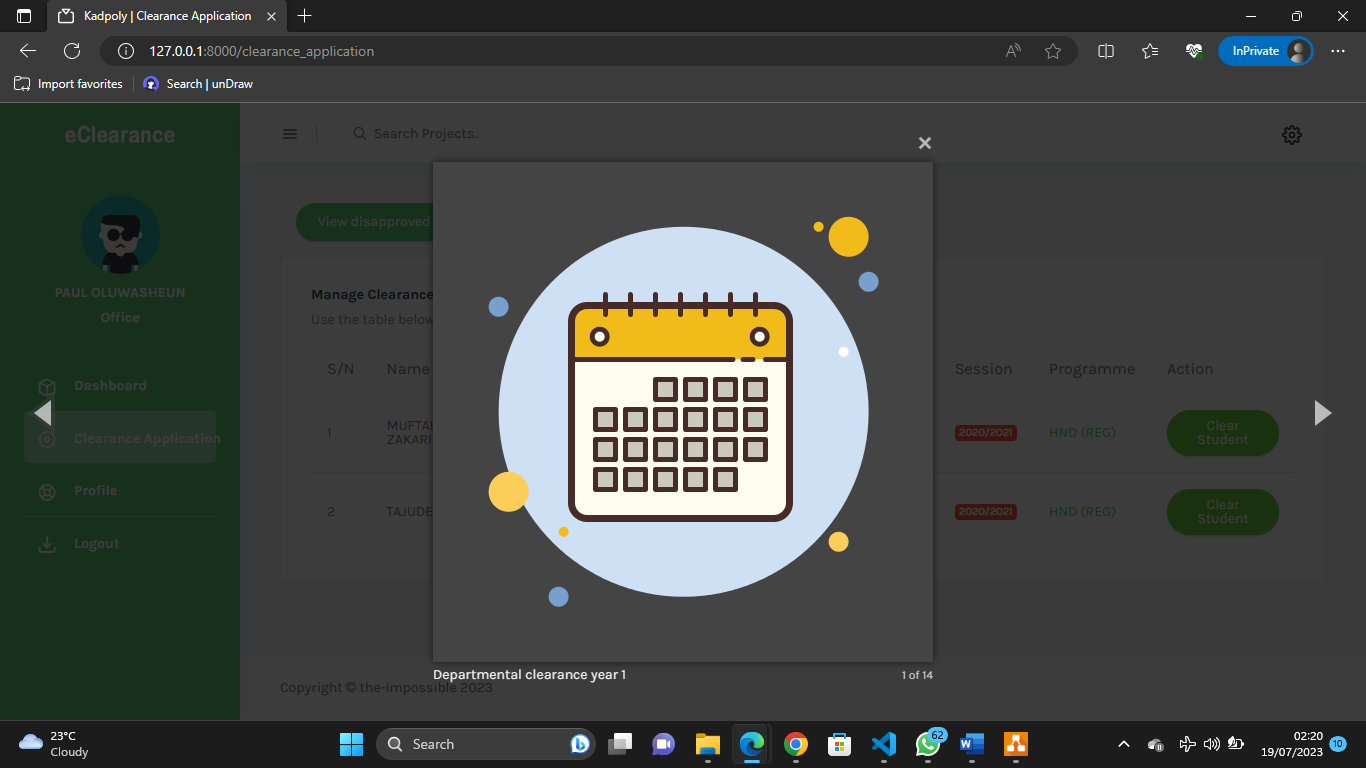


Fig 4.11 View Files

**View Disapproved Clearance**

Administrative office can view all disapproved clearance application

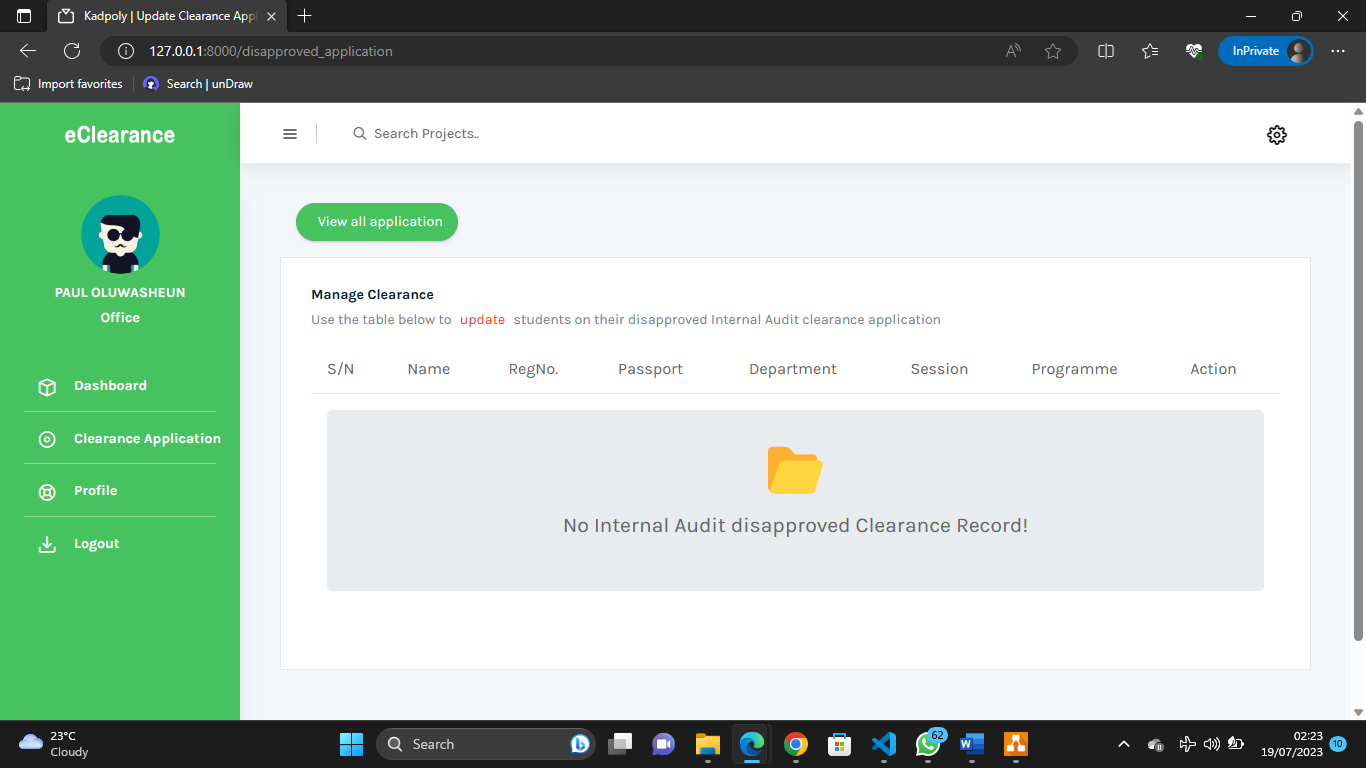


Fig 4.12 View Disapproved Clearance

**Update Administrative Office**

This page enables admin to make changes or even delete administrative office information

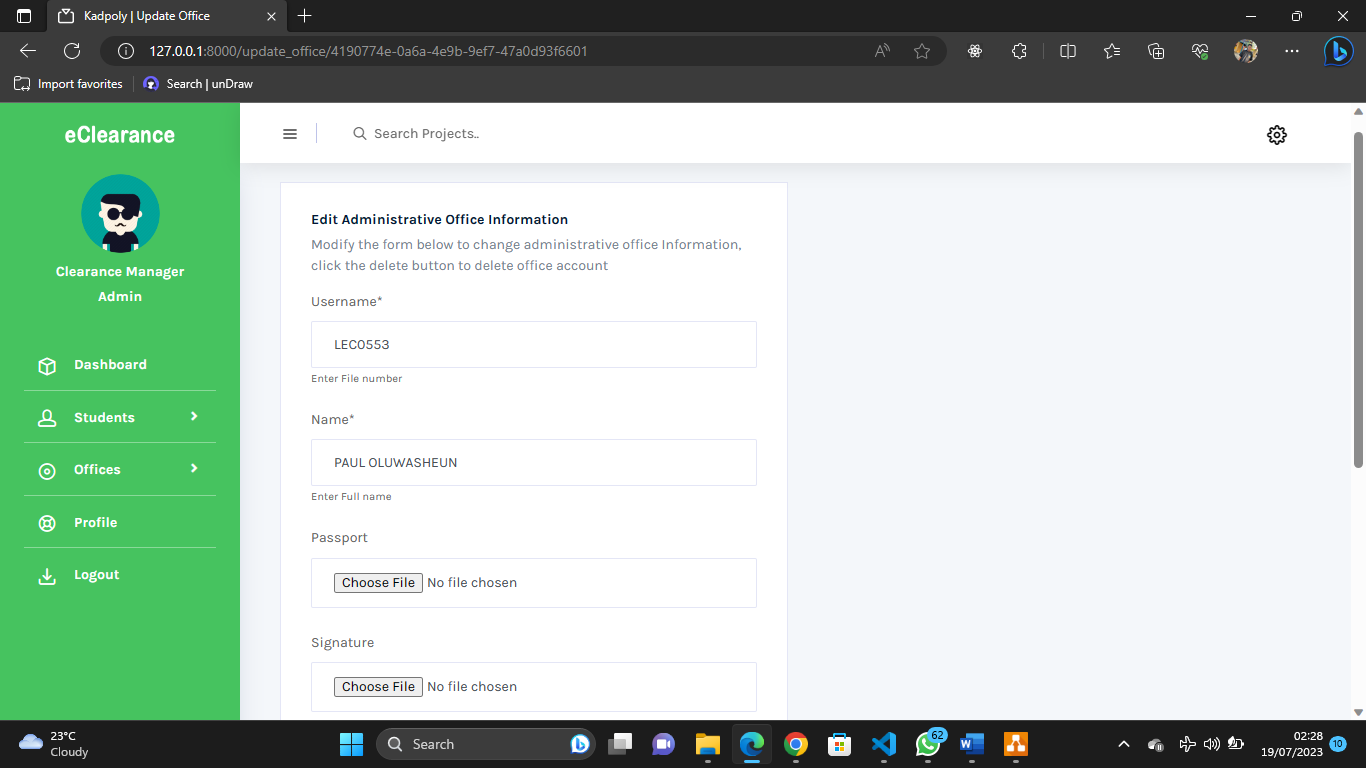


Fig 4.13 Update Administrative Office

**CHAPTER FIVE**

**SUMMARY CONCLUSION AND RECOMMENDATION**

**5.1 Summary**

The study focuses on the development of an online clearance system for graduating students at Kaduna Polytechnic. Currently, the clearance process is manual, time-consuming, and inconvenient. The aim of the study is to address these issues by creating an online system that streamlines the clearance procedure. The objectives include reviewing relevant literature, designing and developing the online system, and evaluating its efficiency. The scope of the study encompasses an examination of current clearance methods and the feasibility of implementing an online system, specifically for graduating students. The study is limited by restricted access to literature and time constraints. The significance of the research lies in the computerization of the clearance process, making it more efficient and accessible from anywhere in the world.

**5.2 Conclusion**

In conclusion, this study focuses on the development of an online clearance system for graduating students at Kaduna Polytechnic. The current manual clearance process is time-consuming and inconvenient, leading to delays and inefficiencies. The aim is to streamline the clearance procedure by creating an online system. The objectives include reviewing literature, designing and developing the system, and evaluating its efficiency. Despite limitations in access to literature and time constraints, implementing an online clearance system will significantly improve the efficiency and productivity of the clearance process. It will also make the process accessible from anywhere, improving record-keeping and enhancing the overall effectiveness of the institution.

**5.2 Recommendation**

Based on the findings of this study, the following recommendations are proposed:

1. Implementation of the Online Clearance System: It is recommended that Kaduna Polytechnic should proceed with the implementation of the developed online clearance system for graduating students. This will significantly improve the efficiency and effectiveness of the clearance process by reducing the time and effort required for clearance activities.
2. Integration of Push Notifications: Push notifications can be used to provide timely updates and notifications to graduating students regarding the status of their clearance process. This can include reminders for pending documents, approval notifications, and other important information. Push notifications will help keep students informed and engaged throughout the clearance process, reducing the need for manual follow-ups and ensuring a smoother and more efficient experience.
3. Training and Awareness: To ensure a smooth transition to the online clearance system, it is crucial to provide comprehensive training to the staff members involved in the clearance process. This will equip them with the necessary skills to effectively use the system and handle any technical issues that may arise. Additionally, conducting awareness campaigns among the students will help them understand the new system and its benefits, ensuring their cooperation during the clearance process.
4. Security Measures: Given the sensitive nature of student clearance information, it is essential to implement robust security measures to protect the data stored in the online system. This includes regular data backups, secure authentication protocols, and encryption of sensitive information. Adequate measures should also be taken to prevent unauthorized access and ensure the privacy of student records.

By implementing these recommendations, Kaduna Polytechnic can establish an efficient and reliable online clearance system that enhances the overall clearance process for graduating students, improving administrative efficiency and student satisfaction.

**REFERENCES**

Agbo-Ajala, O. & Makinde, O. E., (2015). “A Web-Based Database- Driven Students

Clearance System”. *International Journal of Advanced Research in Computer Science and Software Engineering, 5*(7), 65–69.

Jonathan, O., Misra, S., Makinde, F., Damaševičius, R., Maskeliūnas, R., & León, M. (2019).

“Development of Online Clearance System for an Educational Institution”. ICAI. DOI:10.1007/978-3-030-32475-9\_24

Nneji, G.U., Deng, J., Shakher, S.S., Monday, H.N., Agomuo, D., & Dike, I.D. (2018).

“An Improved e-Clearance Management System for Graduating Students in a University Environment*”. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 74*-80. DOI:10.1109/IEMCON.2018.8614875

Mohammed, Ahmed. (2021). Online clearance management system.

Ramadhan, R. N., Manguri, K. H., & Manguri, B. H. (2020). ‘Design and Implementation an

Open-Source Web Based Clearance System for Iraqi Educational Institutions’. *In Proceedings - 2020 2nd Annual International Conference on Information and Sciences, AiCIS 2020 101–105*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/AiCIS51645.2020.00026>.

**APPENDIX**

**View.py**

from django.shortcuts import render, redirect, reverse

from django.http import HttpResponseRedirect, Http404, HttpResponse

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

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

from django.contrib import messages

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

import csv

import io

import codecs

from django.urls import reverse\_lazy

from eClearanceAuth.models import \*

from eClearanceAuth.forms import \*

# Create your views here.

PASSWORD = '12345678'

class HomePageView(TemplateView):

    template\_name = "frontend/index.html"

class DashboardView(LoginRequiredMixin, TemplateView):

    template\_name = "backend/dashboard.html"

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

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

        if self.request.user.user\_type.user\_type == "Admin":

            context["student"] = User.objects.filter(

                user\_type=UserType.objects.get(user\_type='Student')).count()

            context["office"] = User.objects.filter(

                user\_type=UserType.objects.get(user\_type='Office')).count()

            context["completed"] = DepartmentalClearance.objects.filter(

                is\_cleared=True).count()

            context["applied"] = StudentClearance.objects.all().count()

        elif self.request.user.user\_type.user\_type == "Office":

            try:

                profile = AdministrativeProfile.objects.get(

                    user=User.objects.get(username=self.request.user))

                office\_type = profile.office.office\_title

                if office\_type == 'Library':

                    context["approve"] = LibraryClearance.objects.filter(

                        is\_cleared=True).count()

                    context["disapprove"] = LibraryClearance.objects.filter(

                        is\_disapprove=True).count()

                    context["applied"] = LibraryClearance.objects.all().count()

                elif office\_type == 'Hostel':

                    context["approve"] = HostelClearance.objects.filter(

                        is\_cleared=True).count()

                    context["disapprove"] = HostelClearance.objects.filter(

                        is\_disapprove=True).count()

                    context["applied"] = HostelClearance.objects.all().count()

                elif office\_type == 'Sport':

                    context["approve"] = SportClearance.objects.filter(

                        is\_cleared=True).count()

                    context["disapprove"] = SportClearance.objects.filter(

                        is\_disapprove=True).count()

                    context["applied"] = SportClearance.objects.all().count()

                elif office\_type == 'Internal Audit':

                    context["approve"] = InternalAuditClearance.objects.filter(

                        is\_cleared=True).count()

                    context["disapprove"] = InternalAuditClearance.objects.filter(

                        is\_disapprove=True).count()

                    context["applied"] = InternalAuditClearance.objects.all().count()

                elif office\_type == 'Department':

                    context["approve"] = DepartmentalClearance.objects.filter(

                        is\_cleared=True).count()

                    context["disapprove"] = DepartmentalClearance.objects.filter(

                        is\_disapprove=True).count()

                    context["applied"] = DepartmentalClearance.objects.all().count()

            except AdministrativeProfile.DoesNotExist:

                return None

            except User.DoesNotExist:

                return None

        elif self.request.user.user\_type.user\_type == "Student":

            student\_clearance = StudentClearance.objects.filter(

                student=StudentProfile.objects.get(user=self.request.user))

            if student\_clearance:

                if student\_clearance[0].departmental\_clearance.is\_cleared:

                    context["status"] = "Completed"

                else:

                    context["status"] = "In progress"

            else:

                context["status"] = "Has not applied"

        return context

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 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 ManageStudentView(LoginRequiredMixin, ListView):

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

    model = StudentProfile

    form\_class = CreateSingleStudentForm

    second\_form\_class = CreateMultipleStudentForm

    queryset = StudentProfile.objects.all().order\_by('-date\_joined')

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

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

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

        context["form2"] = self.second\_form\_class

        return context

    def post(self, request):

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

        form2 = self.second\_form\_class(request.POST, request.FILES)

        if 'multiple' in request.POST:

            if form2.is\_valid():

                csv\_obj = csv.reader(codecs.iterdecode(

                    request.FILES['file'], 'utf-8'))

                objs = []

                sub\_objs = []

                session = form2.cleaned\_data.get('session')

                programme = form2.cleaned\_data.get('programme')

                department = form2.cleaned\_data.get('department')

                user\_type = UserType.objects.get(user\_type="Student")

                for row in csv\_obj:

                    objs.append(User(

                        username=row[0].upper(), name=row[1], user\_type=user\_type, password=make\_password(PASSWORD)))

                created\_users = User.objects.bulk\_create(objs)

                for user in created\_users:

                    sub\_objs.append(StudentProfile(

                        user=user, department=department, session=session, programme=programme))

                created\_user\_profiles = StudentProfile.objects.bulk\_create(

                    sub\_objs)

                messages.success(request, "Students has been created")

            else:

                messages.error(request, form2.errors.as\_text())

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

                              context={

                                  'form1': self.form\_class,

                                  'form2': form2,

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

                              })

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

        if 'single' in request.POST:

            if form1.is\_valid():

                session = form1.cleaned\_data.get('session')

                programme = form1.cleaned\_data.get('programme')

                department = form1.cleaned\_data.get('department')

                instance = form1.save(commit=False)

                instance.user\_type = UserType.objects.get(user\_type="Student")

                instance.password = make\_password(PASSWORD)

                instance.save()

                StudentProfile.objects.create(

                    user=instance, department=department, session=session, programme=programme)

                messages.success(request, "Student created successfully!")

            else:

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

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

                              context={

                                  'form1': form1,

                                  'form2': self.second\_form\_class,

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

                              })

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

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

class UpdateStudentView(LoginRequiredMixin, SuccessMessageMixin, UpdateView):

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

    form\_class = EditSingleStudentForm

    success\_message = 'Updated Successfully!'

    queryset = User.objects.all()

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

    def form\_valid(self, form):

        student = StudentProfile.objects.get(user=form.instance)

        student.session = form.cleaned\_data.get('session')

        student.department = form.cleaned\_data.get('department')

        student.programme = form.cleaned\_data.get('programme')

        student.save()

        form = super().form\_valid(form)

        return form

class DeleteStudentView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = User

    success\_message = 'Deleted Successfully!'

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

class ManageOfficeView(LoginRequiredMixin, ListView):

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

    model = User

    form\_class = CreateAdministrativeProfileForm

    queryset = AdministrativeProfile.objects.all().order\_by('-date\_created')

    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 form1.is\_valid():

            administrative\_office = form1.cleaned\_data.get(

                'administrative\_office')

            administrative\_office\_department = form1.cleaned\_data.get(

                'administrative\_office\_department')

            if administrative\_office.office\_title == 'Department' and administrative\_office\_department == None:

                form1.add\_error('administrative\_office\_department',

                                'Select administrative\_office\_department')

                messages.error(

                    request, 'Select administrative office department if office is related to a department')

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

                              context={

                                  'form1': form1,

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

                              }

                              )

            if administrative\_office.office\_title != 'Department' and administrative\_office\_department != None:

                administrative\_office\_department = None

            signature = form1.cleaned\_data.get('signature')

            instance = form1.save(commit=False)

            instance.password = make\_password(PASSWORD)

            instance.user\_type = UserType.objects.get(user\_type="Office")

            instance.save()

            AdministrativeProfile.objects.create(

                user=instance, office=administrative\_office, a\_departmental\_office=administrative\_office\_department, signature=signature)

            messages.success(request, "Student created successfully!")

        else:

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

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

                          context={

                              'form1': form1,

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

                          })

        if administrative\_office.office\_title != 'Department' and administrative\_office\_department != None:

            admin.a\_departmental\_office = None

        else:

            admin.a\_departmental\_office = administrative\_office\_department

        admin.office = administrative\_office

        admin.save()

        form = super().form\_valid(form)

        return form

class DeleteAdministrativeOfficeView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = User

    success\_message = 'Deleted Successfully!'

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

class ApplyClearanceView(LoginRequiredMixin, SuccessMessageMixin, TemplateView):

    form\_class = ClearanceForm

    template\_name = "backend/users/apply\_clearance.html"

    success\_message = "Application successful you can now track status"

    def has\_applied(self):

        student\_clearance = StudentClearance.objects.filter(

            student=StudentProfile.objects.get(user=self.request.user))

        if student\_clearance:

            self.form\_class = ClearanceForm(

                instance=student\_clearance[0].internal\_audit\_clearance)

            return [True, student\_clearance[0]]

        return [False, None]

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

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

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

        context["clearance"] = self.has\_applied()[1]

        context["has\_applied"] = self.has\_applied()[0]

        return context

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

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

        if 're-upload' in request.POST:

            student = StudentProfile.objects.get(user=self.request.user)

            clearance = StudentClearance.objects.get(student=student)

            form = self.form\_class(

                request.POST, request.FILES, instance=clearance.internal\_audit\_clearance)

        if form.is\_valid():

            instance = form.save(commit=False)

            student = StudentProfile.objects.get(user=self.request.user)

            passport = str(student.user.passport).split("/")[-1].split(".")[0]

            if passport == 'user':

                messages.error(self.request, "You have to update your profile")

                return render(self.request, self.template\_name, context={'form': form, 'has\_applied': self.has\_applied()[0]})

            if 're-upload' in request.POST:

                instance.is\_disapprove = False

                instance.disapproval\_reason = ""

                instance.save()

                messages.success(self.request, self.success\_message)

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

            else:

                print("NO 3")

                clearance = StudentClearance.objects.create(student=student)

                library = LibraryClearance.objects.create(clearance=clearance)

                hostel = HostelClearance.objects.create(clearance=clearance)

                sport = SportClearance.objects.create(clearance=clearance)

                department = DepartmentalClearance.objects.create(

                    clearance=clearance, department=student.department)

                instance.clearance = clearance

                instance.save()

                clearance.library\_clearance = library

                clearance.hostel\_clearance = hostel

                clearance.sport\_clearance = sport

                clearance.internal\_audit\_clearance = instance

                clearance.departmental\_clearance = department

**Homepage**

{% extends 'base.html' %}

{% load static %}

{% block title %}Homepage{% endblock title %}

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

  {% block body %}

      <!--header section start -->

      <div class="header\_section">

         <div class="container">

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

         </div>

         <!--banner section start -->

         <div class="banner\_section layout\_padding">

            <div id="my\_slider" class="carousel slide" data-ride="carousel">

               <div class="carousel-inner">

                  <div class="carousel-item active">

                     <div class="container">

                        <div class="row">

                           <div class="col-md-6">

                              <div class="taital\_main">

                                 <h4 class="banner\_taital">clearance System</h4>

                                 <p class="banner\_text">an electronic clearance system for graduating students</p>

                                 <div class="read\_bt"><a href="{% url 'auth:login' %}">Click to Sign-In</a></div>

                              </div>

                           </div>

                           <div class="col-md-6">

                              <div><img src="{% static 'frontend/images/img-1.png' %}" class="image\_1"></div>

                           </div>

                        </div>

                     </div>

                  </div>

               </div>

            </div>

         </div>

         <!--banner section end -->

      </div>

      <!--header section end -->

      <!--about section start -->

      <div class="about\_section layout\_padding">

         <div class="container">

            <div class="row">

               <div class="col-md-6">

                  <div class="image\_2"><img src="{% static 'frontend/images/img-2.png' %}"></div>

               </div>

               <div class="col-md-6">

                  <h1 class="about\_taital">About <span class="us\_text">Project</span></h1>

                  <p class="about\_text">The aim of this study is to develop an online clearance system for graduating students at Kaduna polytechnic. This research work would include an investigation of the current methods used in clearance, and an evaluation of the feasibility and potential benefits of implementing an online clearance system. The research work will only be cleared for graduating students and not departmental clearance </p>

                  <div class="read\_bt\_1"><a href="{% url 'auth:login' %}">Get Started</a></div>

               </div>

            </div>

         </div>

      </div>

      <!--about section end -->

      <!--services section start -->

      <div class="services\_section layout\_padding">

         <div class="container">

            <h1 class="service\_taital"><span class="our\_text">Our</span> Objectives</h1>

            <p class="service\_text">The aim of this study is to develop an online clearance system for graduating students at Kaduna polytechnic.  </p>

            <div class="services\_section\_2">

               <div class="row">

                  <div class="col-sm-4">

                     <div class="icon\_1"><img src="{% static 'frontend/images/icon-1.png' %}"></div>

                     <h4 class="design\_text">Objective 1</h4>

                     <p class="lorem\_text">To review the related literature on the online clearance system for graduating students at Kaduna polytechnic </p>

                  </div>

                  <div class="col-sm-4">

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

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

{% endblock body %}