**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**

Getting resources to prepare for exam or general school work is a very important aspect of education and is particularly important for higher institution student where they are in a research environment of which a lot of study has been done earlier in their field of 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).

In preparation for an exam, there are a few things you can do to increase your chances of success. ensure one understands the material that will be covered on the exam. This means going over your notes, attending review sessions, and asking your lecturer for clarification on any topics you don't understand. Second, create a study schedule and stick to it. This will help you ensure you have enough time to review all of the material before the exam. Most importantly, the study and use of past questions by students can be an effective way for them to prepare for an upcoming exam. Past questions provide a great source of insight into the types of questions that may be asked and the topics that will be covered on the exam. By reviewing past questions, students can become familiar with the exam format, the types of questions they may be asked, and the topics they should focus their studying on. In addition, past questions can be used to identify areas of weakness and improve studying strategies (Writer, 2022).

A retrieval system for past exam questions by students will be a web-based platform. This platform could provide the students with access to a database of past questions from various courses and departments. The platform could also enable the students to search, filter, and sort past questions to find the ones they need. In addition, the platform could provide students with the ability to create their own collections of past questions and store them in a personal library. This could help students easily access and review the questions they have already seen.

In the case of this project, the resources that would be provided on a platform would be past questions. Past questions go a long way to help students to prepare for examinations and these resources have been in the past, manually sourced. In this project, a software in form of a website will be developed for an ease of access and retrieval for student.

**1.2 Statement of the Problem**

The success rate of student’s exams depends on how well prepared they were before each exam, it is no doubt that past questions of exams are essential for preparing for exams regardless of the high institution, access to these examinations past questions is actually difficult since it was done before the particular student session. Students have to visit the different business centers or café to seek past questions. It is quite uncertain if they will get the past questions, this task is repeated each semester which obviously the stress involved is cumbersome, or ask students from the previous session who might have misplaced or thrown them away since they are done with the semester which brought about the idea of developing a web-based retrieval system for study past exams question for students to help in boosting student exam success rate.

**1.3 Aim and Objectives of the Study**

To develop a web-based retrieval system for study past exams question for students.

**Objectives**

The objectives of this research work are as follows:

1. To design a working platform where past question papers can be retrieved and stored for posterity reasons.
2. To implement a system where students can find very important resources for them to work with, especially in times of examination preparation.
3. To evaluate how efficiently the system manages the information stored on it.

**1.4 Scope of the Study**

The focus of the study is solely on the development of a web-based retrieval system for study past exams questions for student examination it will only focus on just the main campus of Kaduna polytechnic and it will be limited to few departments on campus, the study will not go beyond this.

**1.5 Limitations of the Study**

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

**Inadequate information –** This factor is the leading limitation of this research work as it put a bar on how far the research work could go.

**Finance –** Some material that would have greatly facilitated the research was restricted to payment before access which prevented the researcher from gaining information that would have been of immense value to this research work

**1.6 Significance of Study**

The study bears significance in the sense that it can help at least better prepare students for examinations, as well as bringing relief for the students by enabling them access all the past examination resources at the comfort of their home and there will no longer be a need to be physically present at the school to fully access and take advantage of its resources. It is an innovative idea that makes full use of the digital age to improve lives of students.

**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. **Past Questions**: A past question is an examination paper from a previous year or previous years, usually used either for exam practice or for tests.
5. **Data Retrieval**: In this context, data retrieval is a process of identifying and extracting data from the database that is based on a query which is provided by the user.
6. **Data Upload**: Data upload refers to the transmission of data from one computer system to another through means of a network.

**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**

Kayode et al. (2021). An android-based blood bank information retrieval system. Over the last decades, blood bank record keeping has been done manually utilizing a paper file management system, which is sluggish for information retrieval and processing and also prone to mistakes in an emergency case. Materials and procedures: This study addresses the aforementioned issue by developing both a web-based and an Android-based blood bank information retrieval system. The online application is used by system administrators at various blood banks to update their available blood inventory information, and the mobile application, which includes a mobile search engine, is used to search for blood supplies from registered blood banks. A component of the system also allows registered blood banks to send a notification to registered blood donors on the application seeking blood donation

Moreso, in recent years, the number of volunteer blood donors has grown in comparison to compensated blood donors. Despite a rise in volunteer blood donors, many people are unable to give blood due to a dearth of knowledge about blood donation. As a result, there have been ongoing losses of acquirable blood from people eager to give blood. Emergency patients in need of blood typically request blood through advertisements on televisions or social media; however, even with a number of advertisements for blood donation, the patient may still not receive the required quantity of blood at that time. This research aims to create an application with all of the necessary features to facilitate contact between blood searchers, blood donors, and blood banks.

In conclusion, the blood center database system was created in response to the need to locate blood supplies or a willing donor in a timely and efficient manner. This method should be made accessible to everyone because it will speed up the search for blood supplies in emergency situations, thereby avoiding health complications and potential deaths caused by delays in the search for blood.

Agboola & Shaibu (2019). The Impact of ICT on information retrieval systems in academic libraries. The purpose of this study was to assess the influence of information and communication technology (ICT) on information retrieval systems in academic libraries. The study was directed by three objectives, three research questions, and four commendations to reach the goal of this research. The survey research design was used for the study, and data was collected using a questionnaire as the instrument. The information gathered was examined using descriptive statistics, frequency counts, and percentages. Education, security, politics, business, infrastructure, and social amenities are all regarded economic resources that boost country development and other resources. This viewpoint is supported by evidence that the possession, manipulation, and use of information and communication technology may improve the cost-effectiveness of many physical and cognitive functions. As a result, modern ICT facilities in academic libraries such as computers, internet, intranet/extranet, local area network (LAN), printers, scanners, machines (photocopy, bindery, and laminating), broadcasting technologies (radio, public address speaker, and television), projector, and telephony, among others, are required for an effective information retrieval system and service delivery

Moreso, to carry out this quantitative investigation, a survey research methodology was used. This study's subjects are library users, including employees, scholars, and students. A total of sixty (60) interviewees were chosen for the research on purpose due to their consistency in using the campus library. The instrument was administered within two (2) weeks to guarantee that those respondents were reached and the instrument was finished. The questionnaire data were evaluated using a frequency count and related percentages.

In conclusion, according to this research, information and communication technology (ICT) has a beneficial effect on academic libraries' information retrieval systems. An information retrieval system is intended to locate and retrieve papers or information needed by the user group. It makes the right information accessible to the right person, and all of this is possible thanks to information and communication technologies.

Gill et al. (2019). Smart system for the retrieval of digital educational content. Education is a significant producer, user, and depositary of instructional information. Today's instructors and students have ubiquitous and on-demand access to knowledge because to technological advancements. We can now connect and exchange information from anywhere in the globe thanks to advances in technology. The availability of huge volumes of diverse educational information, on the other hand, would be useless unless we search, retrieve, and integrate it, therefore building interoperable educational environments. The existing difficulties in integrating instructional information stem from its dispersal across several sources. AIREH (architecture for intelligent retrieval of educational material from heterogeneous settings) is proposed in this study for digital content retrieval using agent-based virtual organizations. Through an information retrieval approach that incorporates both case-based reasoning and federated search, this adaptable architecture supports the search for and integration of diverse material. AIREH is also built on an adaptive organization paradigm for distributed planning, which allows it to handle open systems flexibly, dynamically, and effectively. The case study results are quite encouraging and highlight the benefits of deploying agent-based virtual organizations in the retrieval of tagged digital information. The suggested model is adaptable, adaptable, comprehensive, and efficient.

In conclusion, this study introduced AIREH (framework for Intelligent Retrieval of Educational material in Heterogeneous Environments), a framework for retrieving educational material from partner groups. This paper suggests a retrieval model based on an agent-based virtual organization design to improve digital content search. Multi-agent systems are renowned for their ability to adapt to changes in their environment rapidly and efficiently. The paradigm enables the creation of an open and adaptable framework for dynamic-search-related services for distributed digital material.

**2.3 Summary of Related Literature Reviews**

|  |  |  |
| --- | --- | --- |
| **Author & Year** | **Title & Description** | **Merit and Demerits** |
| Kayode et al. (2021) | An android-based blood bank information retrieval system.  The system will aid in speeding up the search for blood supplies in emergency situations. | The program is both effective and efficient in accomplishing project objectives while also being user-friendly.  A high influx of users might slow system performance. |
| Agboola & Shaibu (2019). | The Impact of ICT on information retrieval systems in academic libraries.  The purpose of this study was to assess the influence of information and communication technology (ICT) on information retrieval systems in academic libraries | ICT has a beneficial effect on academic libraries' information retrieval systems  The researchers suggests that school management should support the library financially to enable them to acquire  all the necessary ICT facilities needed for the information retrieval system |
| Gill et al. (2019). | Smart system for the retrieval of digital educational content.  This paper suggests a retrieval model based on an agent-based virtual organization design to improve digital content search | The paradigm enables the creation of an open and adaptable framework for dynamic-related-search.  The research current challenge is integrating educational content from  several repositories |

**2.4 Problem Inherent in** **the Current System**

The current manual system of retrieving past exam questions can have several inherent problems, including:

1. Time-consuming: Manually searching for past exam questions paper can be a time-consuming process, especially if there are many question papers to search through.
2. Difficulty in organizing and updating question papers: The manual system may not be easy to organize, and updating it with new question papers might be challenging, leading to outdated and irrelevant question papers.
3. Inefficiency in finding relevant question papers: Without a proper search function or indexing system, finding specific past exam question papers within a large database can be difficult.

Overall, the manual system of retrieving past exam questions can be inefficient and time-consuming. It may not be able to provide students with a comprehensive and up-to-date database of past exam questions, leading to suboptimal exam preparation.

**2.5 Analysis of the New Proposed System**

The study proposes a web-based retrieval system of past study exams question for students to prepare for exams by providing them with relevant questions to practice. This can also be beneficial for educators who need to create exams or assess student progress which will also be user-friendly and cost-effective

**2.5.1 Advantages of the Proposed System**

1. Time-saving: The system can save time for students and educators who would otherwise spend significant amounts of time searching for past exam questions manually.
2. Improved exam preparation: By providing students with relevant past questions, the system can improve their exam preparation, helping them to understand the exam format, types of questions, and concepts likely to be tested.
3. Enhanced learning outcomes: With access to relevant past exam questions, students can focus on the areas they need to improve, leading to enhanced learning outcomes and academic performance.

**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 retrieval system for study past exams question for students.

**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**

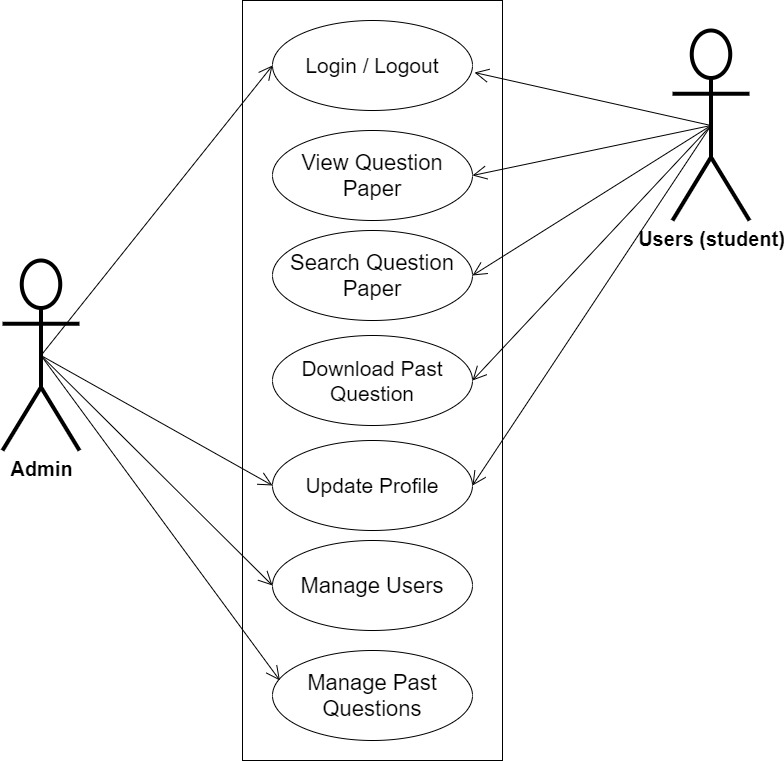
This essentially includes all of the information that someone can receive from existing sources such as books, the internet, case studies, articles, newsletters, and other relevant publications. The resources obtained from the internet in particular were quite relevant; various search engines, particularly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

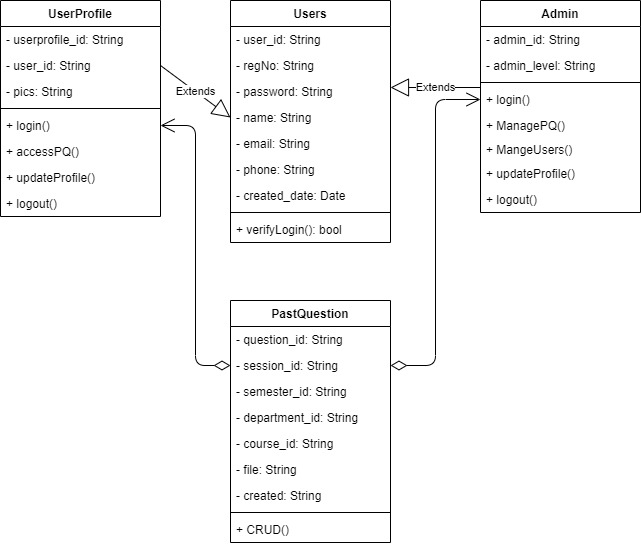
Use cases are groups of interactions between a system and a user. Use case diagrams are used to graphically portray the functioning of a system in terms of its actors, goals (expressed as use cases), and dependencies between those use cases.



**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).



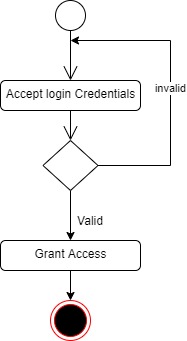
**Fig 3.2 System Class Diagram**

**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.

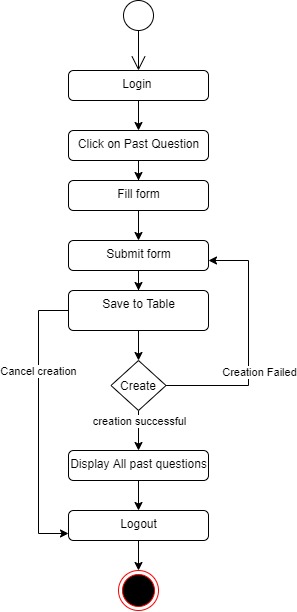
**Login**

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**

**Uploading Past Question**

The process for uploading a past question paper is depicted below, to upload a question paper one has to be authenticated.

**Fig 3.3.2 Upload Past Question 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. Past Question Table: contains every system-saved past question 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 Past Question Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| question\_id | Varchar | No | PK | 32 | Unique string identifying notice |
| session\_id | Varchar | No | FK | 32 | Reference to an academic session |
| semester\_id | Varchar | No |  | 255 | Reference to an academic semester |
| Department\_id | Varchar | No |  | 255 | Reference to a department |
| course\_id | Varchar | No |  | 20 | Reference to a course |
| File | Varchar | No |  | 100 | File uploaded |
| created | Date | No |  | 10 | Date the past question was uploaded |

**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**

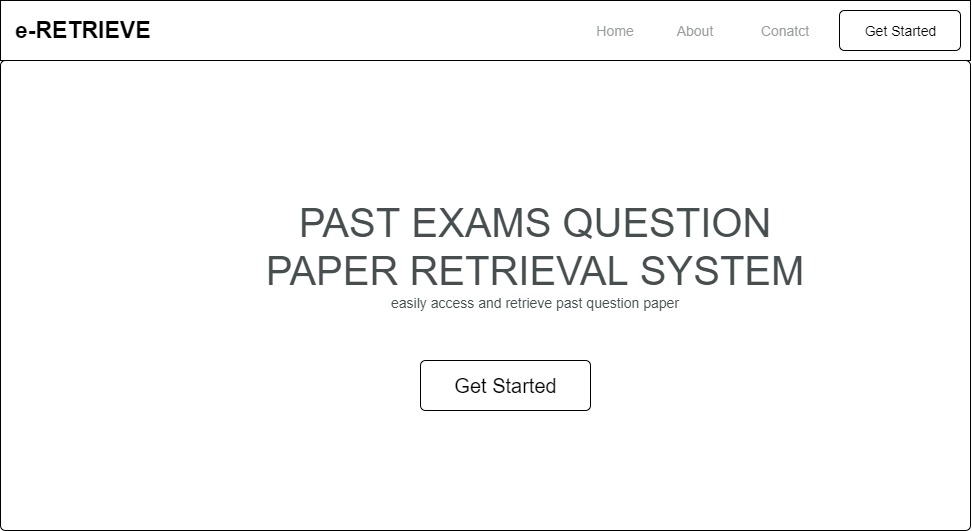
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User\_id** | **regNo** | **Name** | **Email** | **Phone** | **Created\_date** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX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

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

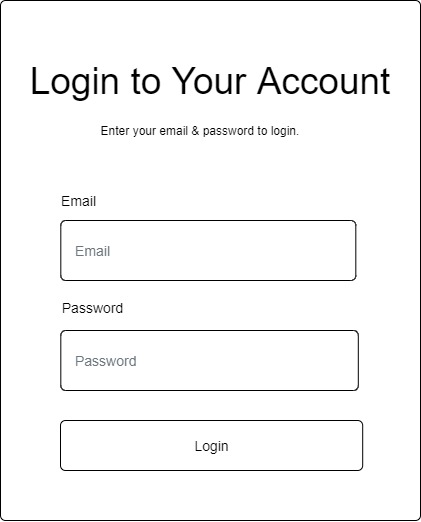
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **question\_id** | **session\_id** | **semester\_id** | **department\_id** | **course\_id** | **file** | **created** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | 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**

**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 research work will be a mobile-based application and will be implemented on a relational database system (SQLite). HTML, CSS, 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**

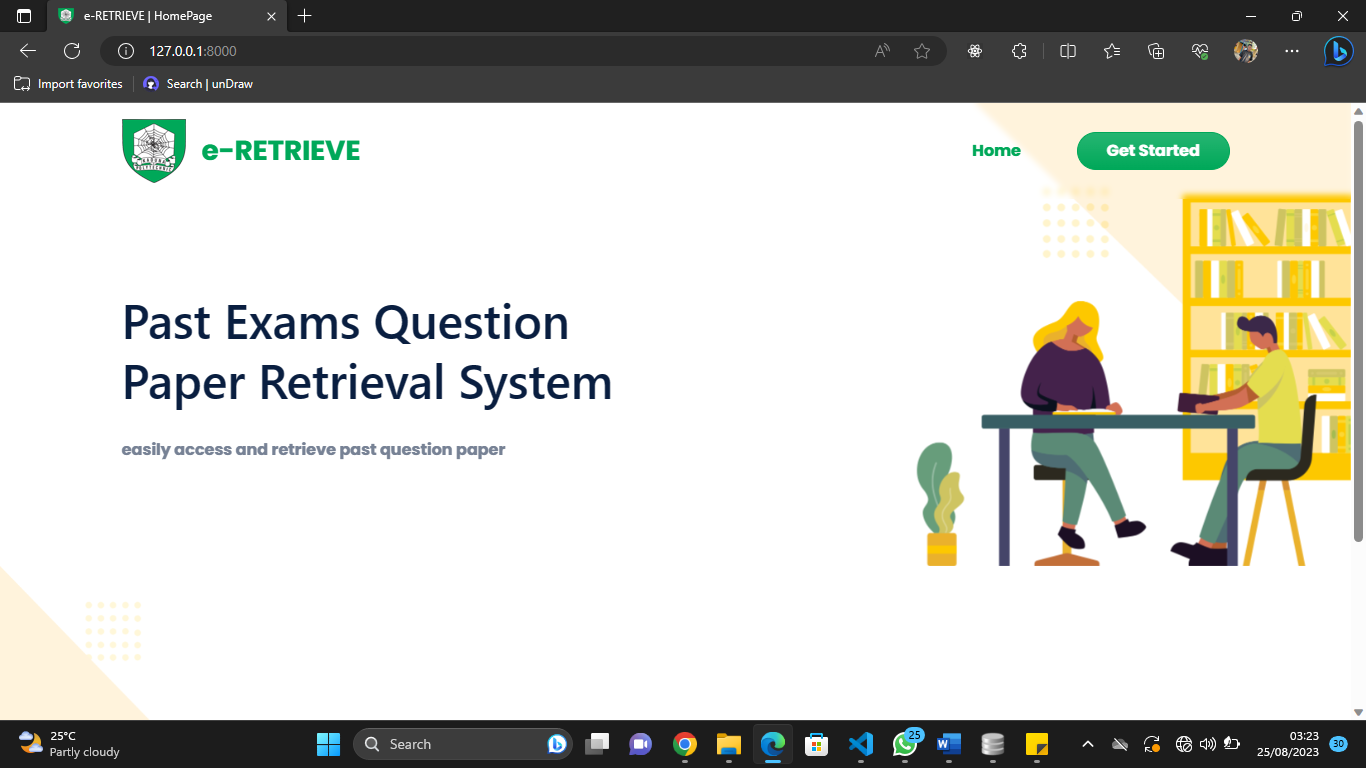
Given that the website has a public scope, certain information such as the index page and login page are accessible to all visitors. However, there are specific functionalities and information that are restricted and not available to every visitor. To enforce these restrictions, passwords are implemented, 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 and student) access to the system only if the correct credentials are provided

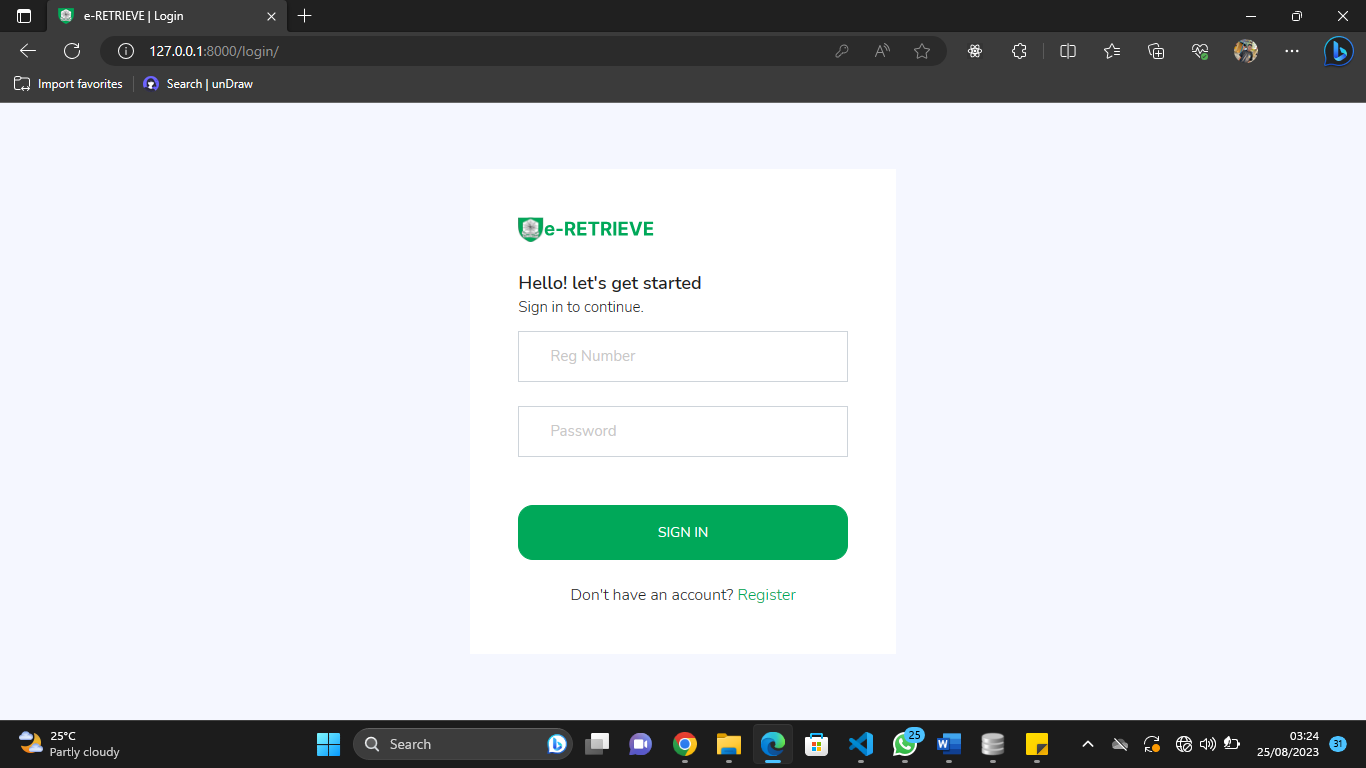


Fig 4.2 User Login

**User Registration**

This is a page that students makes use of when creating an account, in order to gain access to the system

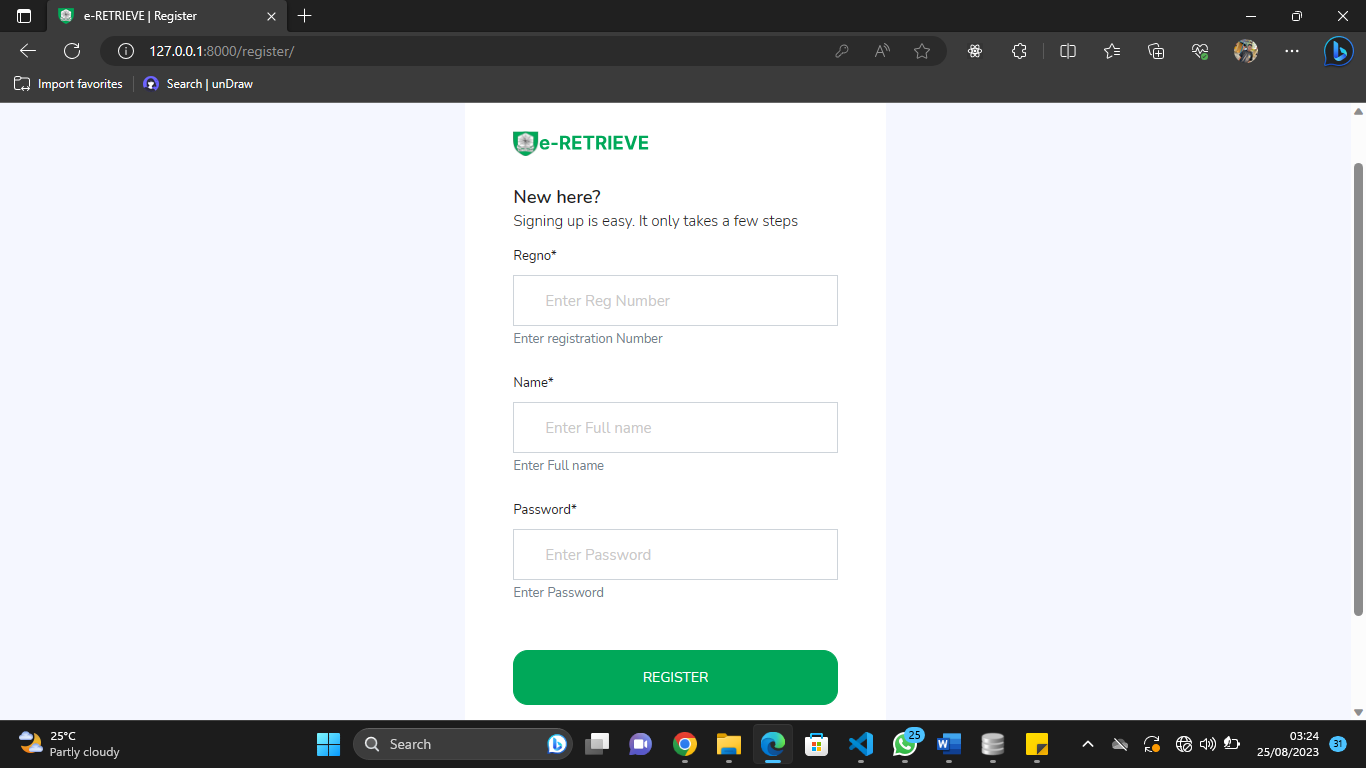


Fig 4.3 User Registration

**Admin Dashboard**

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

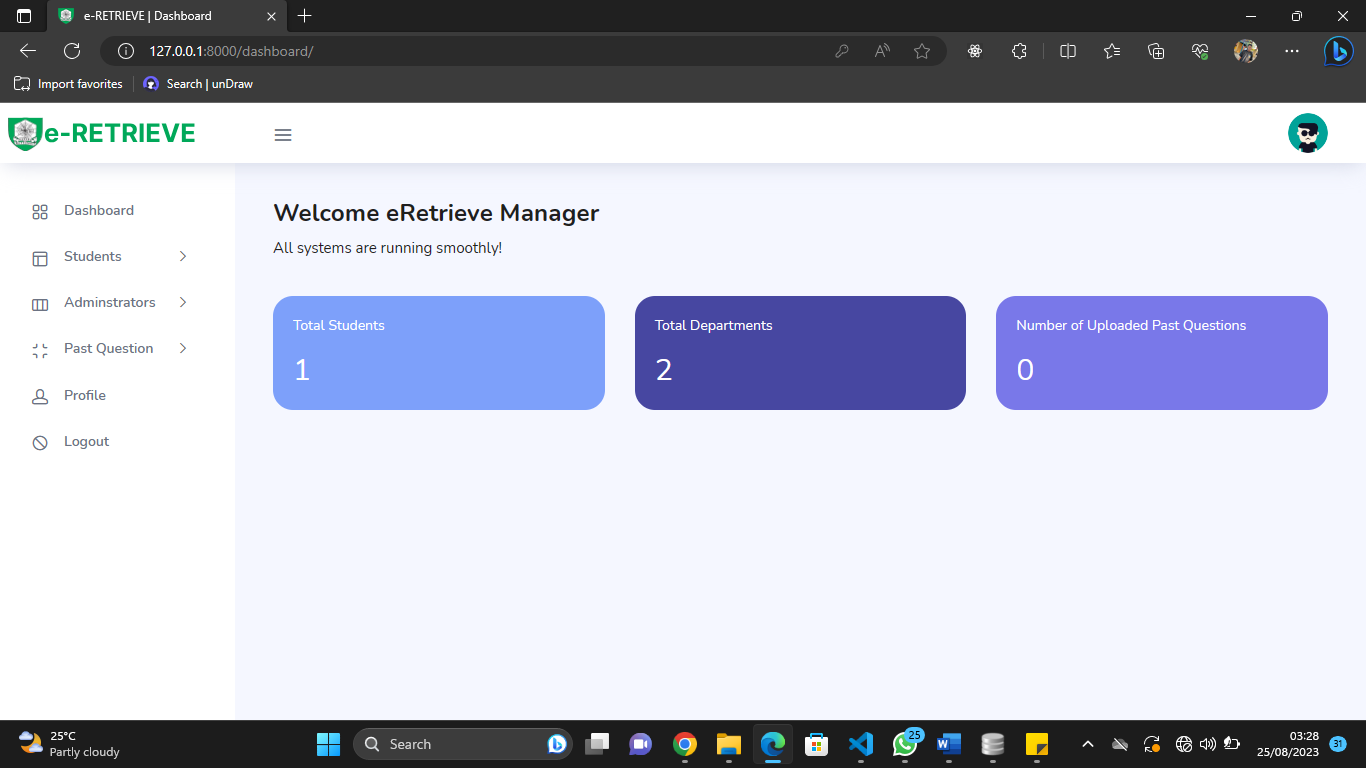


Fig 4.4 Admin Dashboard

**Manage Student’s Account**

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

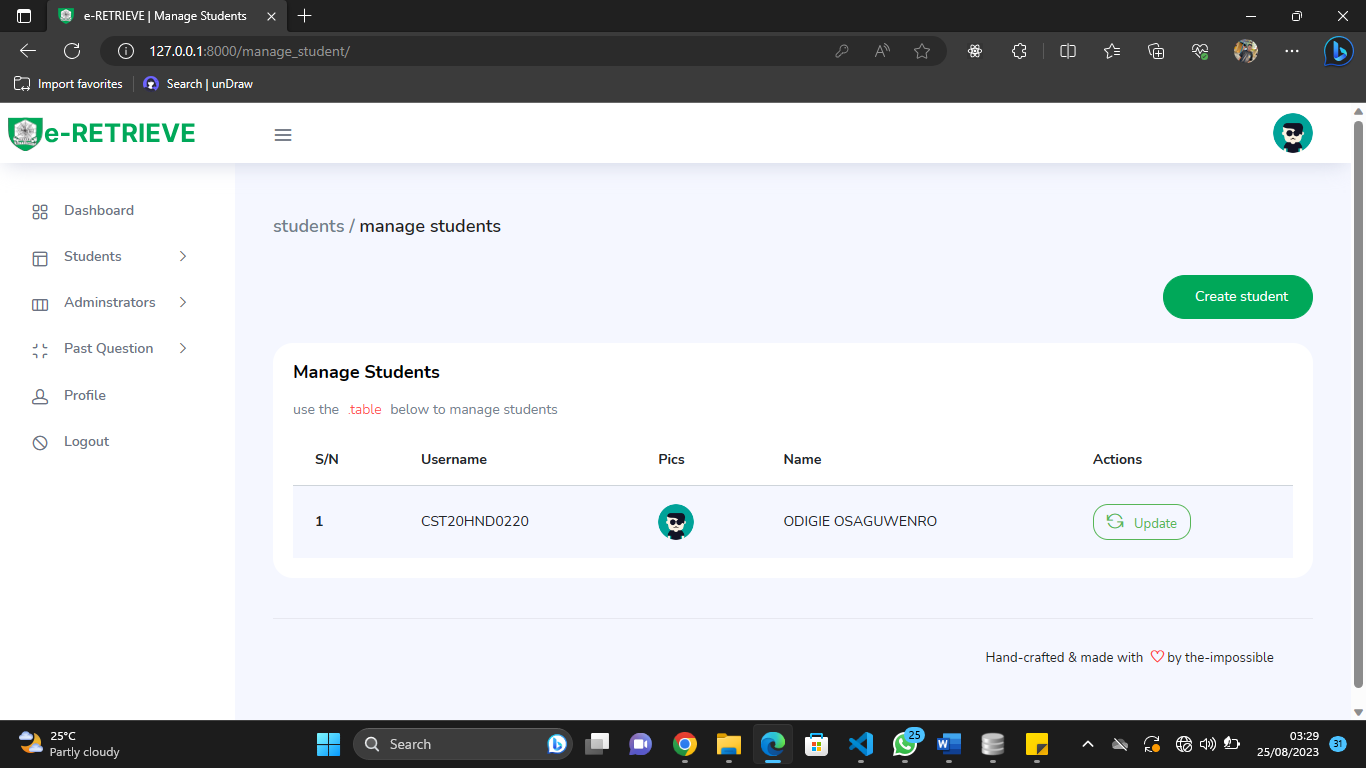


Fig 4.5 Manage Student’s Account

**Manage Administrative**

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

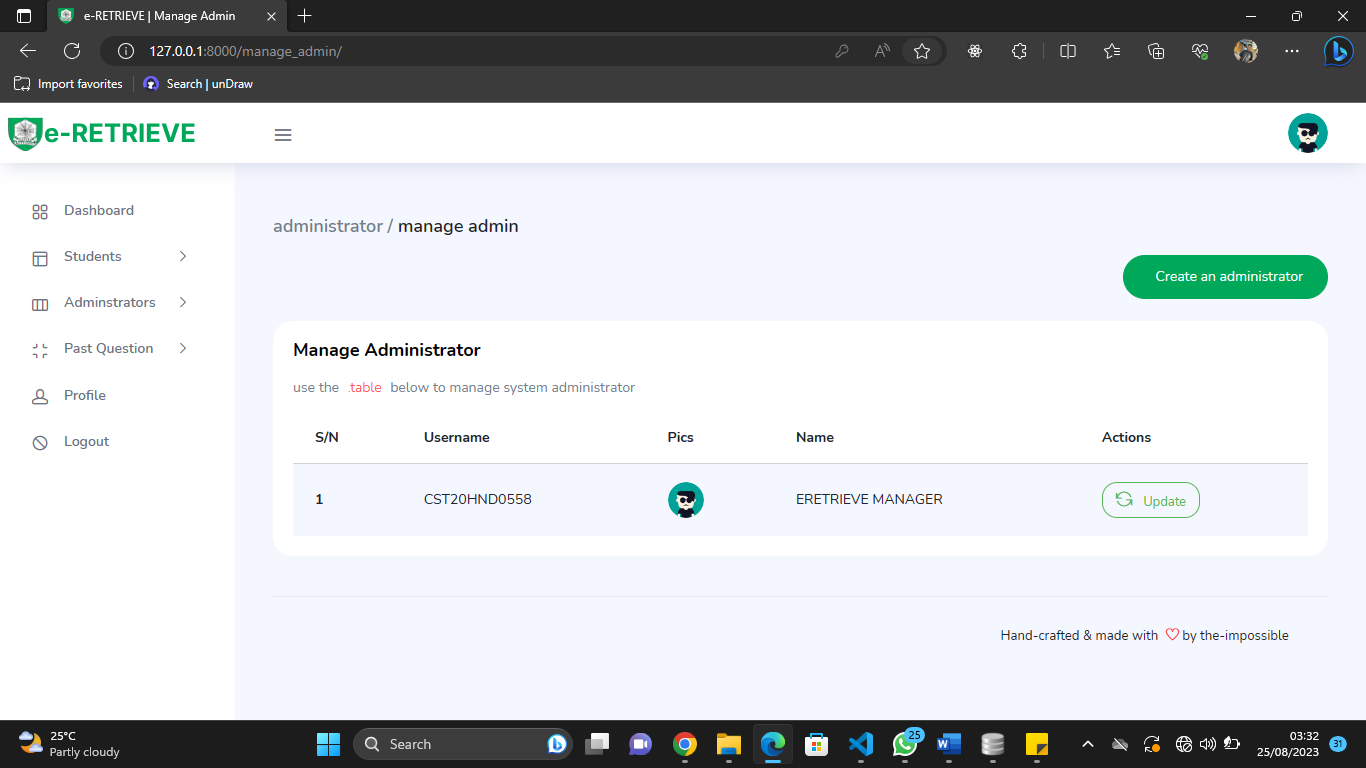


Fig 4.6 Manage Administrative Officers

**Upload Past Question**

The admin can upload past question based on the department, semester and session for student to download

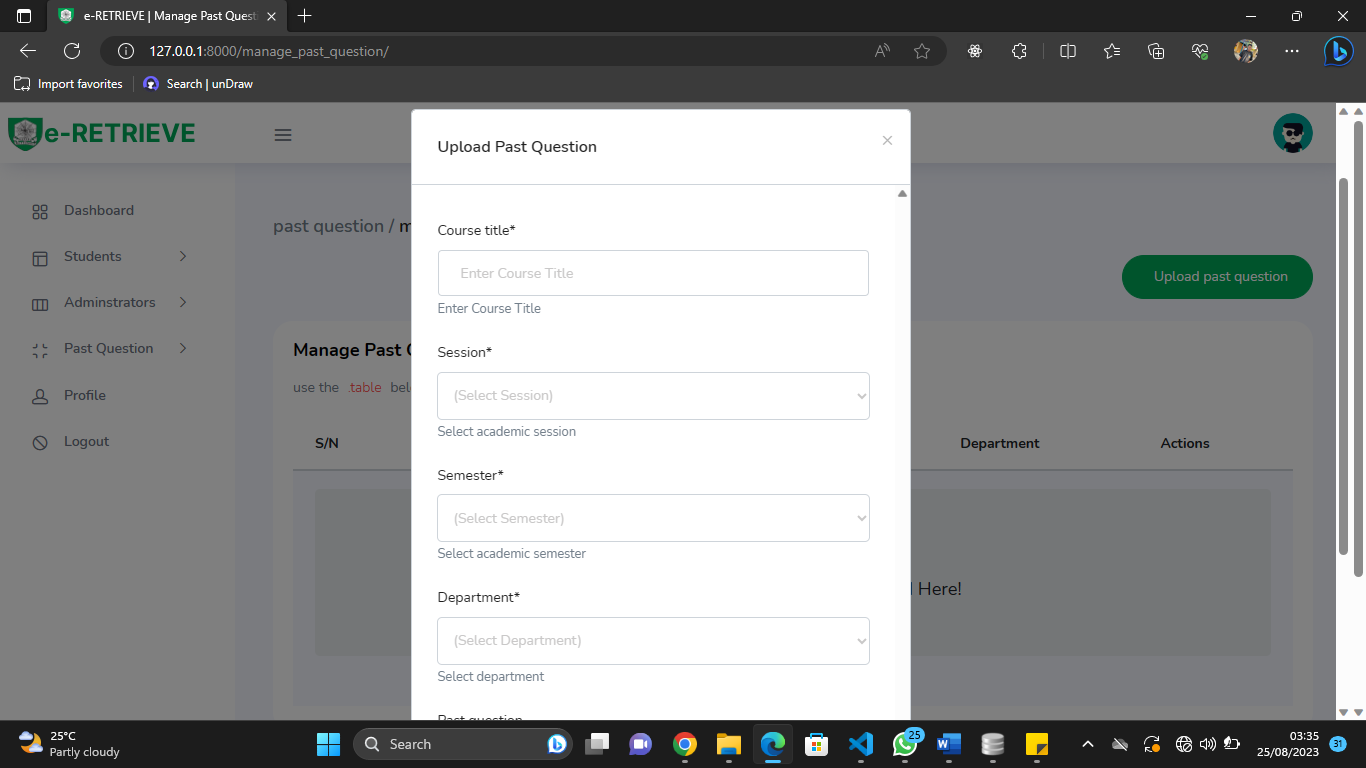


Fig 4.7 Upload Past Question

**Manage Past Question**

This is the page where the admin can effectively manage all the uploaded past questions

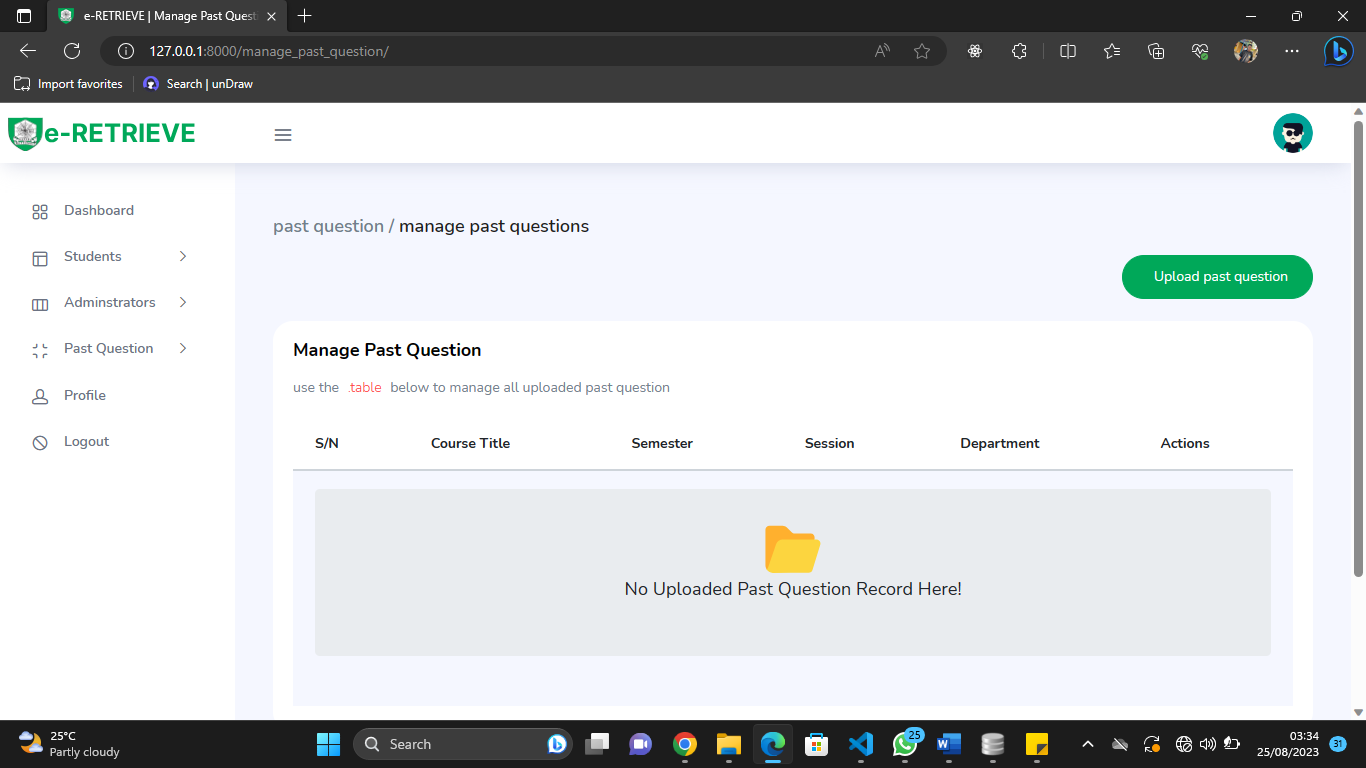


Fig 4.8 Manage Past Question

**Account Profile**

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

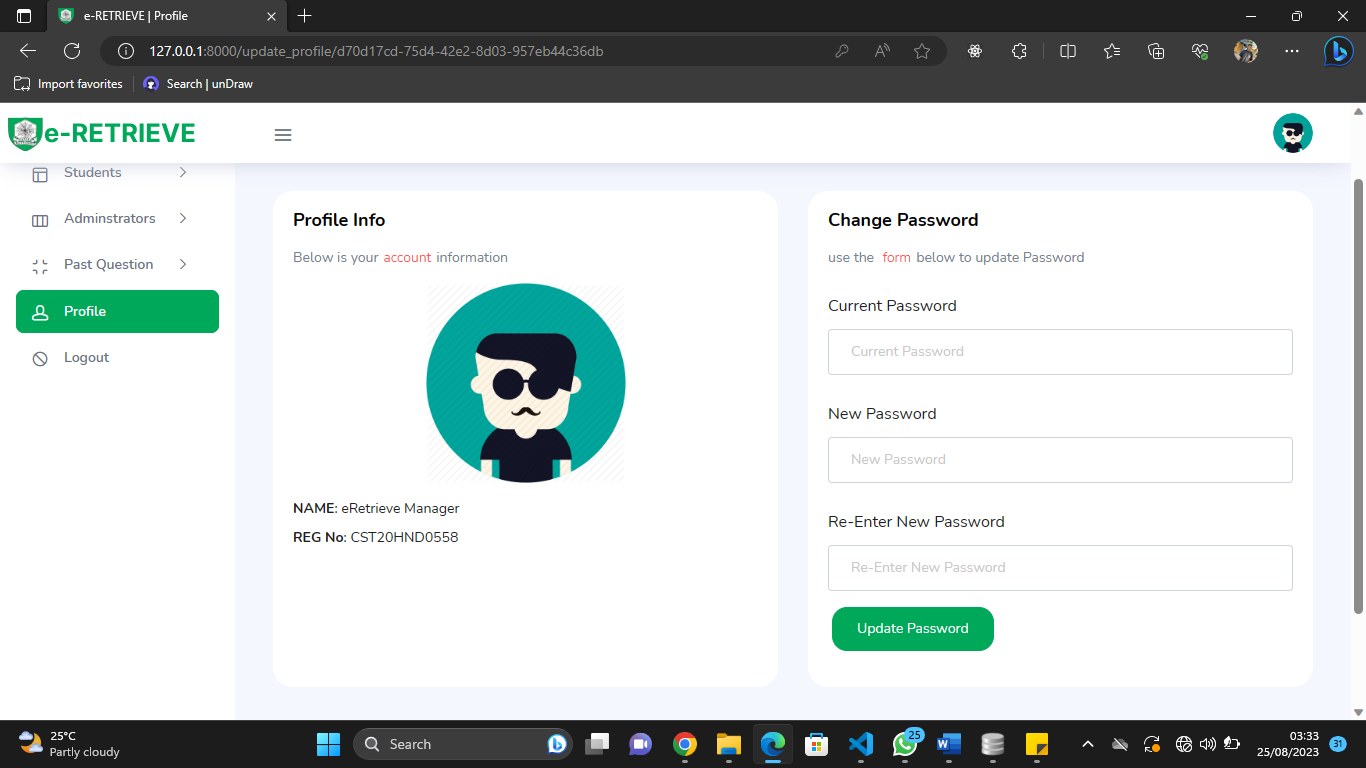


Fig 4.9 Manage Account Profile

**Student Dashboard**

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

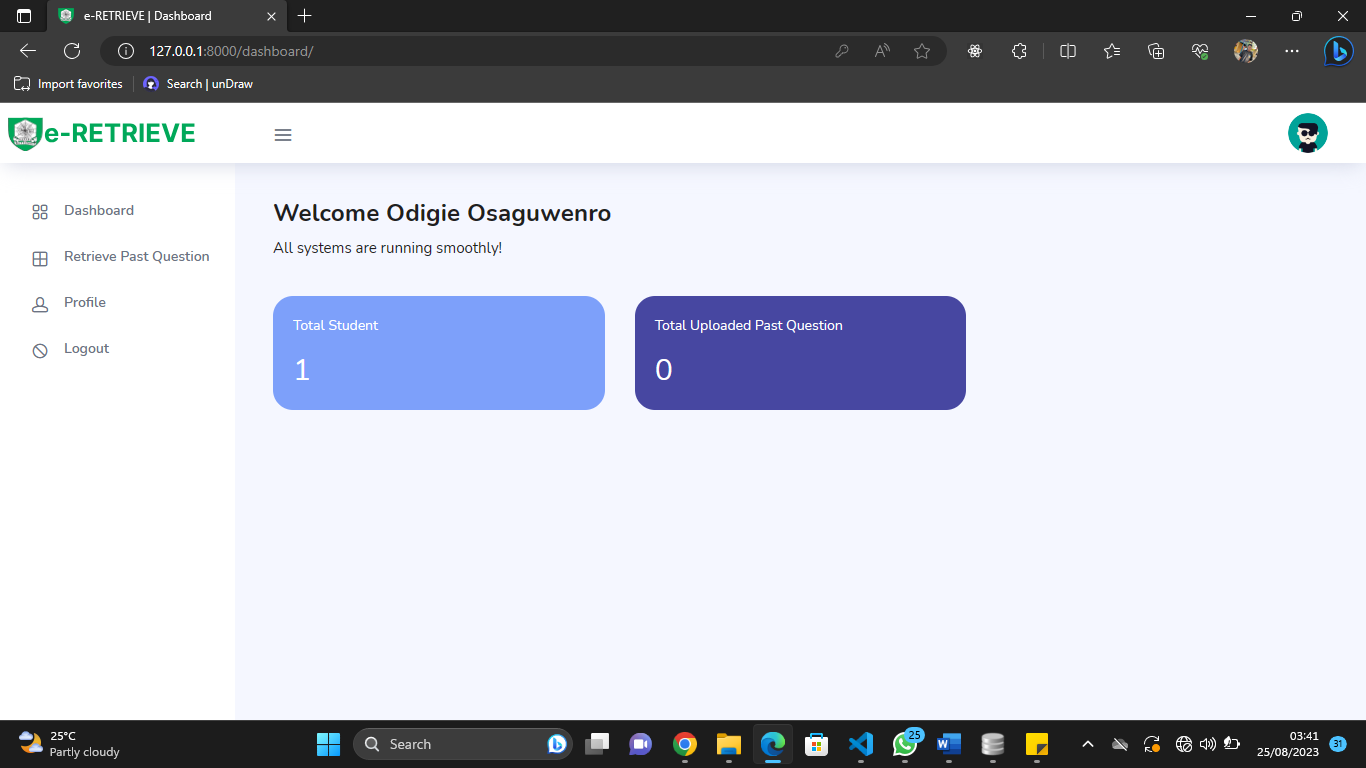


Fig 4.10 Student Dashboard

**Past Question Retrieval Page**

The student can use this page to retrieve and download uploaded past questions

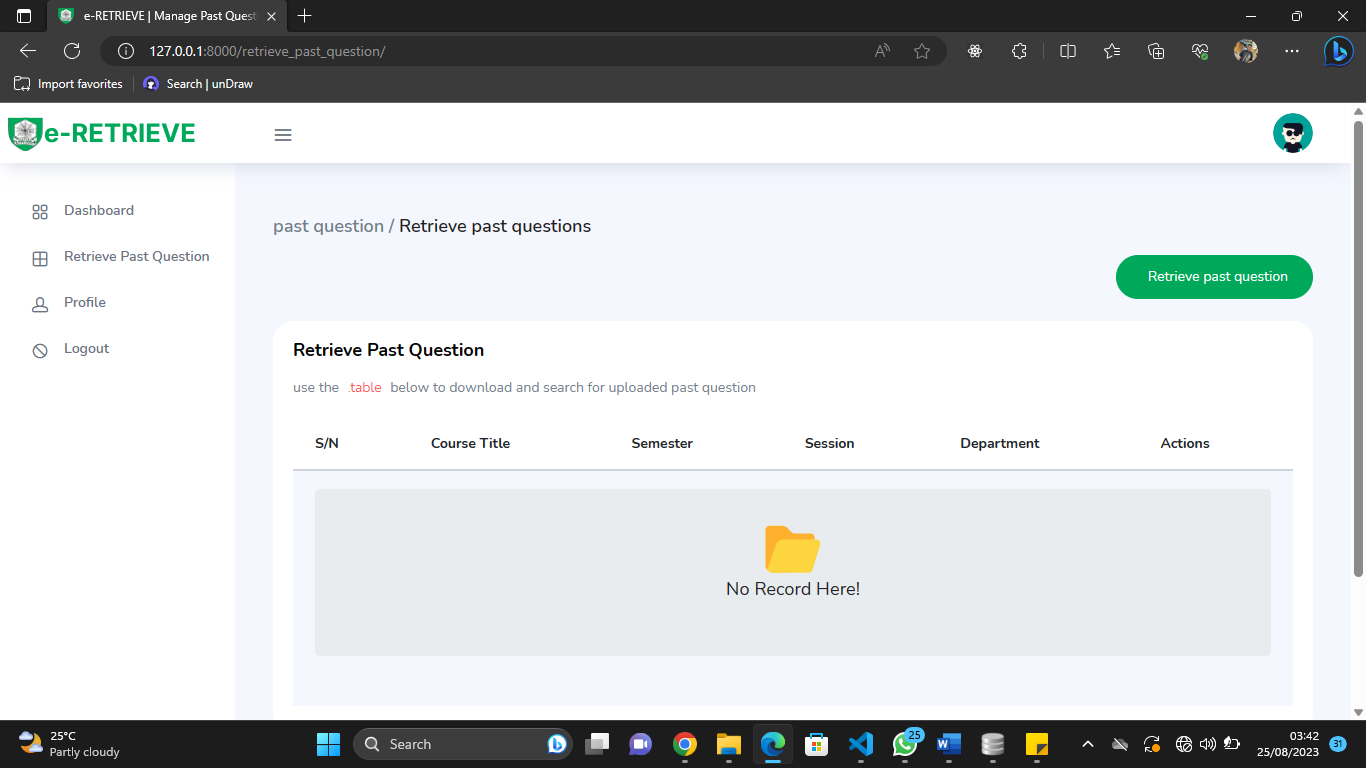


Fig 4.11 Past Question Retrieval Page

**CHAPTER FIVE**

**SUMMARY CONCLUSION AND RECOMMENDATION**

**5.1 Summary**

In the realm of education, the accessibility of resources for exam preparation is paramount, particularly in higher institutions where students engage in research-intensive environments. Examinations serve as vital assessments of a student's knowledge and skills, with various formats such as written tests, oral exams, or practical demonstrations. Adequate exam preparation is essential for gaining a deep understanding of the subject matter.

This study acknowledges the importance of exam preparation and emphasizes the value of utilizing past questions as a valuable resource for students. Past questions offer insights into potential exam formats, types of questions, and crucial topics. They also aid in identifying areas of weakness and enhancing study strategies. To address the challenges associated with obtaining past exam questions, this project aims to develop a web-based platform. This platform will grant students access to a comprehensive database of past questions from diverse courses and departments. It will empower students to search, filter, and organize past questions according to their needs. Furthermore, students can create personalized collections of past questions for easy access and review.

**5.2 Conclusion**

In conclusion, effective exam preparation is pivotal for students' success. Past questions have long been recognized as a valuable resource for exam readiness. However, the process of obtaining these resources has been cumbersome and uncertain, often involving visits to business centers or reliance on the goodwill of students from previous sessions. This study has identified this challenge and proposed a solution in the form of a web-based retrieval system for past exam questions.

The development of such a system holds promise in alleviating the stress associated with acquiring past questions. It provides students with a convenient, digital platform to access, store, and organize past exam questions. By bridging this resource gap, it can significantly contribute to improving students' exam success rates.

* 1. **Recommendation**

Based on the objectives and scope of this study, the following recommendations emerge:

1. **System Implementation**: The development of the web-based retrieval system should proceed efficiently. The system should be user-friendly, ensuring that students can easily access and retrieve past exam questions.
2. **Scaling**: While the initial focus is on Kaduna Polytechnic's main campus and specific departments, there is potential for expansion. Consideration should be given to scaling the platform to cover more campuses and departments in the future.
3. **User Training**: Students should receive adequate training on how to navigate and utilize the platform effectively. User guides and tutorials can be provided to facilitate this process.
4. **Feedback Mechanism**: Implement a feedback mechanism within the platform to allow students to report any issues or suggest improvements. This will help in continuously enhancing the system's functionality.

In essence, the development of this web-based retrieval system represents an innovative step forward in leveraging technology to empower students and enhance their educational experiences. Its successful implementation has the potential to transform exam preparation, making it more accessible and efficient for students.

**REFERENCES**

Agboola, B., & Shaibu, R. (2019). “Impact of ICT on information retrieval system in academic

libraries: The experience of Federal University Gashua Library, Yobe State, Nigeria. Library Philosophy and Practice”.

Gil, A. B., de la Prieta, F., Rodríguez, S., & Corchado, J. M. (2019). “Smart system for the

retrieval of digital educational content”. *Applied Sciences (Switzerland), 9*(20). https://doi.org/10.3390/app9204400

Kayode, A. A., Adeniyi, A. E., Oluwaseun, R., Ogundokun, & Ochigbo, S. A. (2019).

“An android based blood bank information retrieval system”. *Journal of Blood Medicine, 10*, 119–125. <https://doi.org/10.2147/JBM.S197350>

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

Writer, S. (2022, September 20). Exam Preparation: Ten Study Tips. Top Universities. https://www.topuniversities.com/student-info/health-and-support/exam-preparation-ten-study-tips

**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")