CHAPTER ONE

INTRODUCTION

This chapter reviews the description of the project and some related background information on the project. Besides that, this chapter observes the problem statement of the project to give a clear insight of what is the scope and objectives of the project. This chapter will become the guideline for all the work to be carried out in the later stage.

1.1 Background of the Study

Information and Communication Technology is one of the diverse industries in the world that is increasing development in a certain country. Verification is the process of establishing the truth, accuracy, or validity of something such as the verification of official documents (Musee, 2017).

Most of the applicants falsify their educational credentials. What's more, industry experts cite academic fraud as the most common lie on resumes. This poses the greatest danger to an organization. This has been accelerated by applicants who falsify the information. The risks involved in not verifying the applicant's certificate details include, greater recruiting and replacement costs, increased employee turnover, compromised business performance, embarrassment, and a negative impact on Kaduna polytechnic reputation.

In Africa, as time goes by the rate of growth of ICT technology increases from time to time though in Africa, most of the institutions and organizations rely on use traditional paper records verification methods to verify the documents presented to them. These organizations/Institutions do not have the enough capacity to verify the documents presented to them instantly. One of the problems we have in traditional paper based is that people and especially recruiters and employers find difficult in knowing the validity of documents such as academic certificates presented to them because there is no way they can authenticate those documents instantly. In the current scenario most of the organization does not have the capacity to instantly authenticate the documents presented. Traditional identity information verification and validation processes were developed in a human/paper transaction world. In East Africa, Uganda is having a great progress in implementing IT systems than the other countries (Musee, 2018).

Therefore, for this case, a computerized system to obtain graduates certificates records and verification should be introduced in the school which will enable several recruiters to verify the certificate records from the ones issued in the system and the ones that they have.

1.2 Statement of the Problem

Recently employers have been experiencing has high alarming rate of fake certificates (Taylor, 2017). This is due to the traditional paper-based prototype of verification. The issue of forgery of printed certificates is one that is frequently encountered. Similarly, the low skill level required to counterfeit academic certificates is a major issue. The difficulty of the traditional verification process has resulted in unnecessary delays as well as inconvenient long-distance travel for verification purposes. However, with advancements in information and computer technology, a paradigm shift from traditional based verification to real-time verification is required.

1.3 Aim and Objectives of the Study

The project is aimed at designing a working platform in Kaduna polytechnic which will be used nationally and perhaps internationally in verifying the records of academic certificates for all graduates from Kaduna polytechnic in Nigeria.

Objectives

The objectives of this research work are as follows:

i. To design a system that would verify all Kaduna Polytechnic certificate records making sure that they are all valid and original.

ii. To implement a system that can eliminate forgery of certificates.

iii. To evaluate and efficiently manage results information of the students.

1.4 Scope of the Study

This project work is centered on developing a website (Kadpoly e-Verify) with the intention of eliminating the use of fake Kaduna polytechnic results in Nigeria and perhaps internationally. This result verification software reduces instances of academic fraud, also the system reduces the process of sending people to advocates to verify their certificate credentials. The people that are expected to fully utilize this portal are the Student Affairs of Kaduna polytechnic and Recruiting organization. The study will not cover the verification and validation of other school certificates

1.5 Limitations of the Study

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

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

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

1.6 Significance of Study

This study will have a potential impact on Kaduna Polytechnic as it would create a platform for registering each student’s result to the database upon when the need for verification of the result. As well as recruiting organizations and employers will have easy and fast opportunities to verify the certificates presented to them thus it would help them in verifying academic certificates held by an individual.

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

Website: also written as web site, collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server.

Hypertext Markup Language (HTML): HTML is a text-based approach to describing how content contained within an HTML file is structured. This markup tells a web browser how to display the text, images, and other forms of multimedia on a webpage.

Python: Python modern scripting language and interpreter that is freely available and used on general computers.

Common Gateway Interface (CGI): This is a standard way for a Web server to pass a Web user's request to an application program and to receive data back to forward to the user.

Django: Django is a free and open-source cross-platform web server solution stack package that encourages pragmatic design.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The purpose of this chapter is to show the connection between what is presented for study with the existing knowledge, previous studies, or contemporary practice with relevant citation of other scholars work with respect to the problem.

2.2 Literature Review

Clement (2018). Academic Certificate Record Verifying Platform. Employers have been witnessing a highly alarming incidence of phony certificates as a result of the traditional paper-based verification method. This has made it difficult for them to determine the legitimacy of a certificate provided to them, as well as manage the certificates' records, particularly in learning institutions such as schools. As a result, recruiting organizations face far-reaching consequences such as increased recruiting and replacement costs, increased employee turnover, compromised business performance, embarrassment and negative impact on an organization's reputation, declining market value, lost customers and revenue, and civil and criminal liability.

Furthermore, the Agile Methodology approach was used, and ACRVP was built with Laravel PHP Artisan Framework and Bootstrap 4. Taylor Otwell invented Laravel, a free, open-source PHP web framework designed for the building of web applications using the model-view-controller architectural paradigm. Laravel aims to make development easier by simplifying common tasks found in most web projects, such as: robust background job processing, simple, fast routing engine, powerful dependency injection container, database agnostic schema migrations, multiple back-ends for session and cache storage, and real-time event broadcasting.

In conclusion, the researchers can clearly assert that if the certificate records verification procedure and its application were taken seriously internationally, it would undoubtedly contribute significantly to the acceptance of this unique technology. My expectation is that after the entire system has been built and deployed, the difficulties of having phony certificates and their records would be considerably decreased.

Emele et al. (2020). An Enhanced Web Base Certificate Verification System. Certificate verification is a big problem in organizations, educational institutions, recruiters, and employers. These issues were inevitably found after extensive research and analysis of the current system.

i. The manual technique of certificate verification always has a time lag.
ii. Organizations and businesses do not have rapid and simple access to the current system.
iii. Because certification verification takes time, it might be difficult to determine the legitimacy of an academic credential.
iv. Service Defect

Moreso, research methodology outlines what research is, how it is carried out, how progress is measured, and what constitutes success. The technique used in this study is the object-oriented and design methodology (OOADM) while the programming language used for the front-end design was Hypertext Markup Language (HTML5), Cascading Style Sheet (CSS3), Bootstrap Framework and Back end design was achieved with Hypertext Preprocessor (PHP5) and My-Structural Query Language (MySQL).

In conclusion, the enhanced web-based certificate verification systems will be a helpful research tool that will not only open up new approaches for evaluating and validating certificates before accepting them but will also foster uniqueness and trust in companies. With the full implementation of this new system, corporate organizations, individuals, academic institutions, and the government will be able to check any certificate granted to individuals without first consulting the institution that provided it. Once registered with the entity that controls this software, the institution may be easily confirmed.

Patrick et al. (2019). A Generic Certificate Verification System for Nigerian Universities. Certificates are issued by institutions to people who have fulfilled the prerequisites for graduation. However, because of the availability of modern and low-cost scanning and printing technology, certificate forgery has risen, threatening the integrity of both the certificate holder and the university that issued the certificate. As a result, document validation and verification have become critical responsibilities.

Moreso, the system stores certificates in a NoSQL database (MongoDB), and the front-end design is done in PHP. The three parties engaged in the suggested approach to achieve correct certificate verification are the university, the graduate, and the verifier. Using the presented paradigm can provide several advantages. These advantages include enhanced work processes, simplicity of use and maintenance by the University for the Verification Process, and a longer operational duration owing to the usage of MongoDB (a NoSQL database that permits even horizontal scaling).

Finally, this online certificate-checking method prevents academic certificate forgery. It provides less cost involvement and convenience to both developers and users, i.e. significant ease of use by employers because they can obtain original certificates from schools easily and quickly, as opposed to other traditional methods of verification such as manual method, QR code, Watermark, facial recognition, and biometric technology, which are more expensive.

Dinesh et al. (2020). Educational Certificate Verification System Using Blockchain. After the interview procedure is over, the employer takes a long time to provide an offer letter. The employer must authenticate the certificate from the certificate issuing body in order to verify its authenticity. The employer spends a significant amount of time verifying the authenticity of the certificate. To finish the selection procedure, the whole certificate verification process takes longer. To address this issue, Blockchain provides a verified distributed ledger with a cryptographic technique to combat academic certificate forgery.

Futhermore, the system's method for issuing digital certificates is as follows. The first step is to produce the certificate hash value using double SHA256. In the block, save the fixed length hash value as a transaction. This transaction is validated by blockchain members; if it is accepted as a legitimate transaction, the block is added to the current blockchain. The consensus algorithm will be used to accept and reject proposals. The consensus algorithm can be chosen based on the number of nodes and the number of transactions. The system will create the corresponding QR code and inquiry string code to include in the hardcopy certificate. The system includes a device for authenticating hardcopy certificates through phone scanner or online.

Finally, the key characteristics of blockchain applications are transparency and data immutability. It is a distributed ledger in which nodes in the network check and reach a final consensus before adding data to the network. The process of generating academic certificates is open and dispersed among parties, and any organization or party may use this blockchain system to check the information of any academic certificate. academic institutions can work with other employers to publish credentials on the blockchain in order to eliminate false educational certificates.

Thua and Khoa (2019). Eunicert: Ethereum Based Digital Certificate Verification System. Certificates are currently used to assess individuals' knowledge and abilities in a vast and global job market. Unfortunately, this has exacerbated the counterfeiting problem, not only in developing nations but also as a really global one. Each country has a certificate authority that has been approved by respected organizations. These organizations, however, are unable to deliver reliable findings that include both objective and subjective factors. Thousands of colleges and educational institutions throughout the world do not exist, but hundreds of millions of phony degrees can be issued. Verifying the credibility of qualifications is a significant concern for today's businesses.

Moreso, to access web services, users interface with the system via the EUniCert Frontend. The EUniCert Backend links to the EUniCoin system to issue certificates into the EUniCoin Network, which generates transactions and stores them in blocks. EUniCert Frontend supports all procedures such as issuing, obtaining, and validating certificates.

In conclusion, we suggested the EUniCert to enhance the efficiency of transaction verification in the digital issuing and validating system on the blockchain platform based on the findings of our previous study and the better consensus algorithm on the Ethereum platform. We also created new classes in the enhanced system (EUniCert) and established a way to compare performance to the prior system (UniCert).

2.3 Summary of Related Literature Reviews

Author & Year	Title & Description	Merit and Demerits
Clement (2018).	Academic Certificate Record Verifying Platform. This research work is an effort toward the elimination of fake certificates in learning institutions	The system lowers the danger of certificates being lost accidentally or stolen as a result of transferring them on occasion. The system is limited to Tanzania only.

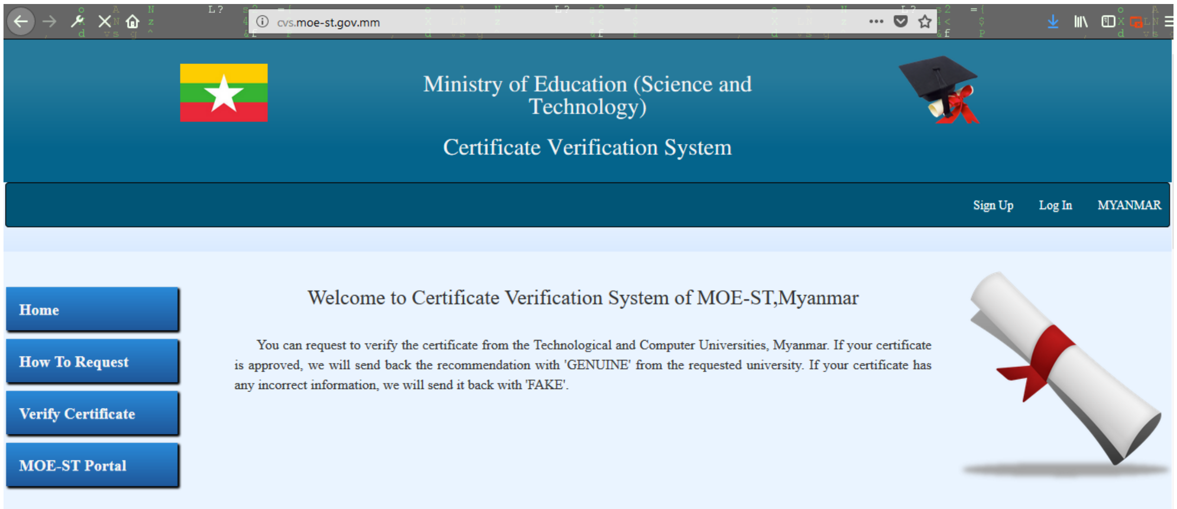
Emele et al. (2020).	An Enhanced Web Base Certificate Verification System. The enhanced web-based certificate verification system will help schools and cooperating organizations validate the authenticity of students' certificates by showing certificate data and format with the owner's image	The system is able to verify and authenticate students' certificates with ease The system may not be scalable due to the choice of programming language used.
Patrick et al. (2019).	A Generic Certificate Verification System for Nigerian Universities. The study aims to develop an online certificate verification system that easily confirms the authenticity of a certificate by employers or recruiters	Employers can obtain original certificates from schools easily and quickly. The study is limited to information gathered from the review of the literature
Dinesh et al. (2020).	Educational Certificate Verification System Using Blockchain. The system uses blockchain technology to offers a verified distributed ledger with a cryptographic	The system ensures transparency and data immutability Lot of computing power is required.

	technique to combat academic certificate forgery.	
Thua and Khoa (2019).	Eunicert: Ethereum Based Digital Certificate Verification System. The study proposes a solution issuing and verifying digital certificates called EUniCert to solve this problem	The system improved the latency to validate result digitally. Smart contracts were not used to generate digital certificates, which would have increased security and transparency.

2.4 Analysis of the Existing System

Certificate Verification System of MOE-ST, Myanmar.

The Colombian government portal is integrated with Ministry of Education which consist of the certificate verifying site where members can sign up/register, then after successfully login a user can upload their certificate, within a period of one week the user can obtain the results in their account with a status either ‘real’ or ‘fake’. The site has a simple and user-friendly interface which everyone can easily integrate with the system. Figure 3 shows the home interface of Certificate Verification System (CVS) of MOE (Ministry of Education) in The Republic of Union of Myanmar.



Welcome to Certificate Verification System of MOE-ST,Myanmar

You can request to verify the certificate from the Technological and Computer Universities, Myanmar. If your certificate is approved, we will send back the recommendation with 'GENUINE' from the requested university. If your certificate has any incorrect information, we will send it back with 'FAKE'.

Figure 2.1: Home interface of Certificate Verification System (CVS)

Benefits of CVS:

i. It is simple to use.

ii. It is highly beneficial for Myanmar resident.

Protests of CVS:

i. Only verifies resident's certificates and take about a week to verify certificate.

ii. Poor security as anyone can register even if he/she is from a different country.

iCredify India

iCredify is dedicated to fulfill our increasing group of clients by offering them with accurate and quick education verification solutions of the utmost standard that are marks verification, year of passing the exam verification, certificate fraud detection and college enrollment verification. This will bring greater awareness of unseen risk of fraud certificate holders and highlight the essential significance of accurately hiring the right qualified professionals to company. iCredify submits complete data about individual's school and college education records for the companies, a potent report that make easy for companies to recruit. iCredify collects information about academic credentials of job seeker or scholarship applicant.

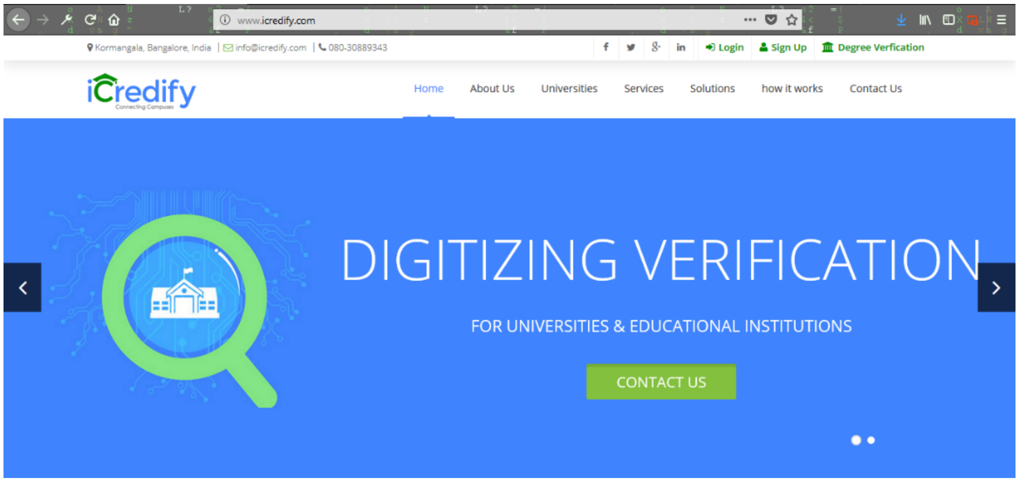


Figure 2.4 iCredify homepage.

Benefits of iCredify

i. Offers attractive user interface that integrates with campus authentication and student records system.

ii. Offers the marks card, verification of percentages of marks and grades obtained in graduation.

Challenges

i. It is not user friendly as it has a lot of options in processes of verification.

ii. It is intended to be used worldwide however it is mostly used in India.

2.5 The Proposed System

There are several existing systems that verify certificate such as the iCredify, and CVS, this new system (Kadpoly e-Verify) is still different from the others because it is more user friendly, multi-dimensional, organizations/recruiters can verify certificates using the same system also the issuing institution can manage the certificates they offer systematically unlike other systems. Therefore because of the large number of job applicants who make false educational claims (Taylor, 2017), certificate records verifications are a valuable honesty check, while helping to protect an employer against negligent hiring claims, hence this is an honourable chance to implement such a system in Kaduna polytechnic so as to get rid of the issue of acquiring fake certificates records from several applicants/employees and even leaders qualifying for a position in the Government.

CHAPTER THREE

METHODOLOGY AND DESIGN

3.1 Introduction

Methodology is a way of careful study or investigation especially in order to discover new fact or information, hence research methodology should be sound enough to make attainment of the set objectives possible with specific components such as methods of data collection and design. This chapter entails the System Modelling (Use case, activity and class diagrams) as well as the input/output specifications and system requirement for the design of this Kadpoly e-Verify portal.

3.2 Method of Data Collection

In executing any system, one has to have an insight of what is happening, it is important that information and fact about the existing system is gathered. In executing this research, two methods were employed

i. Observation of the Work Environment.

ii. Documentation

3.2.1 Observation of the Work Environment

This method was employed in gathering information/data for this research by looking at the way in which the manual system was carried out. The glaring issues with the existing system were detected by careful observation Utilizing the observational method can exert varying amounts of control over the environment in which the observation takes place.

3.2.2 Documentation

The Documentation method is a secondary method of data collection. This method involves the use of journals, handbooks, past projects and newspapers. This method of data collection is used because it serves as a basis of reference to existing research work. This includes internet which is a method of data collection the web was used in sourcing for information on areas that seems difficult or confusing in order to achive an alternative, workable result verification system (Kadpoly e-Verify).

3.3 System Modeling

System model is the conceptual model as a result of the system modelling that describe and represent a system. A system describes a relationship between any set of components to achieve some common objective. The UML applied in this new design include Use Case Diagram, Class Diagram and Activity Diagrams. In this research work, a Unified Modelling Language (UML) is used.

3.3.1 Use Case Diagram

The purpose of a Use Case Diagram is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

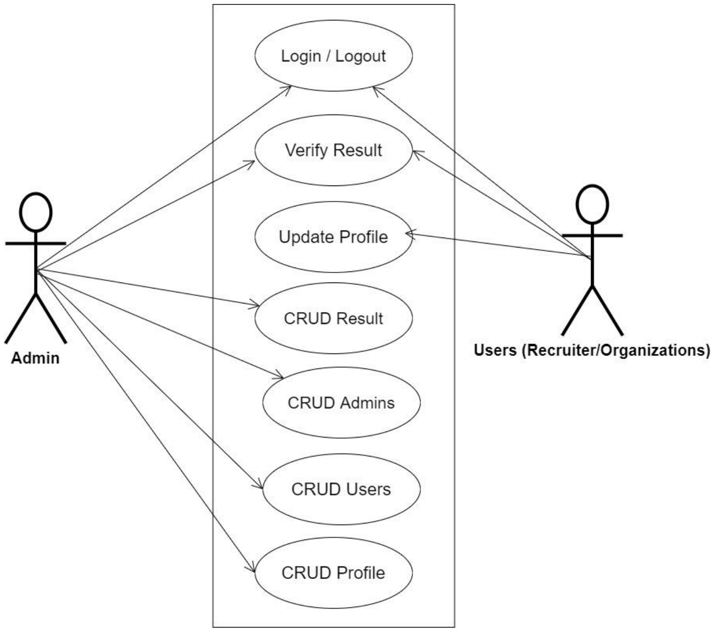


Fig 3.1 System Use Case Diagram

3.3.2 Class Diagram

Class diagram in the Unified Modeling Language (UML) is an implementation of independent view of how the system interface will be, each class with its own attributes and how they are related to each other. Class diagrams are visual representations of the static structure and composition of a particular system using the conventions set by the Unified Modeling Language (UML).

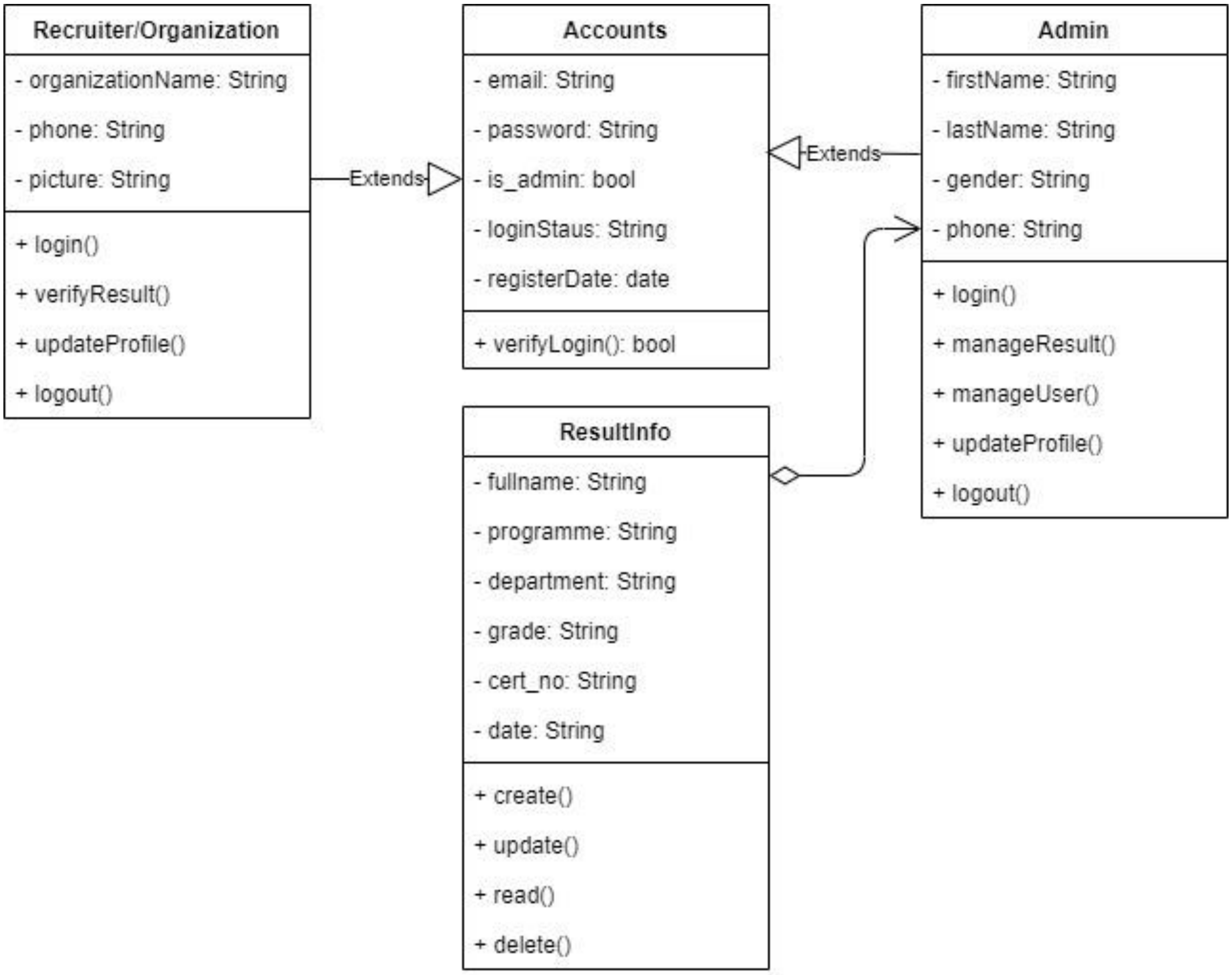


Fig 3.2 System Class Diagram

3.3.3 Activity Diagram

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 username and password must be accurate to gain access.

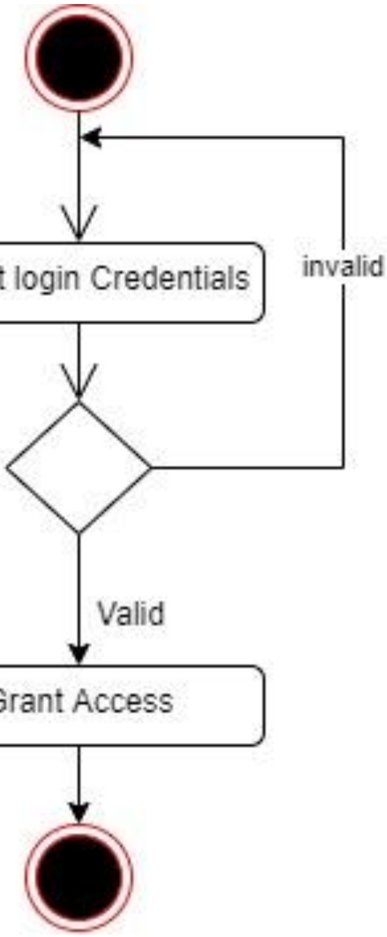


Fig 3.3 Login Activity Diagram

Verify Result

The process for verifying result is depicted in the diagram below; the user must be authenticated to verify result.

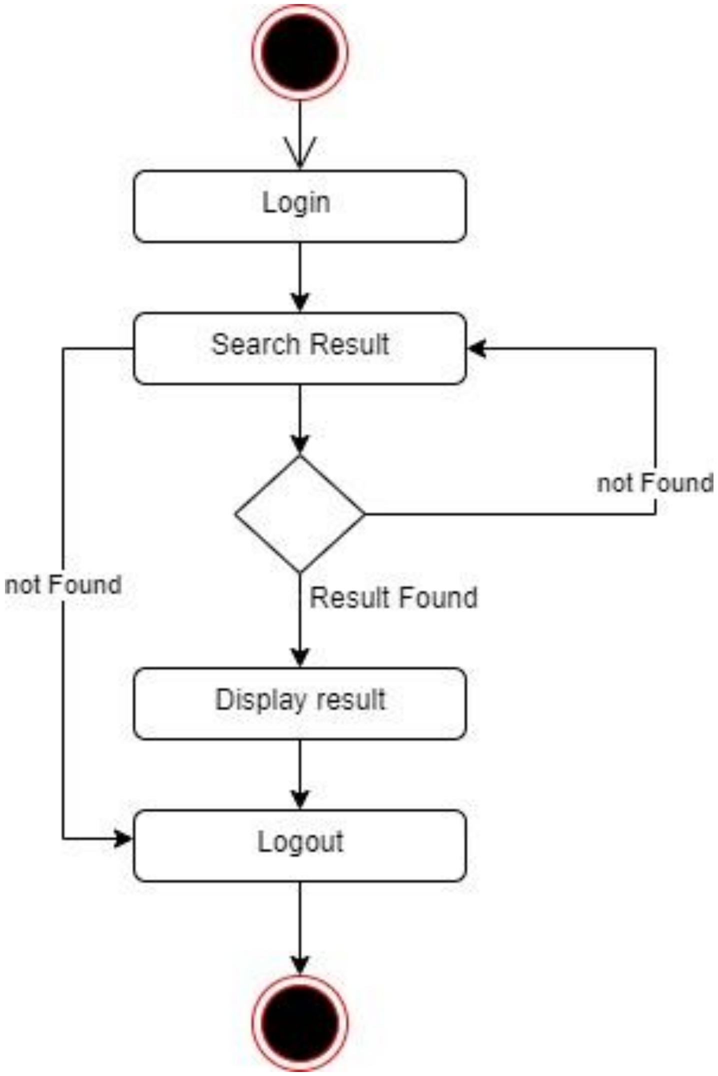


Fig 3.4 Verify Result Activity Diagram

Register Result

The process for registering result is depicted in the diagram below; the user must be authenticated and authorized to perform the registration of result.

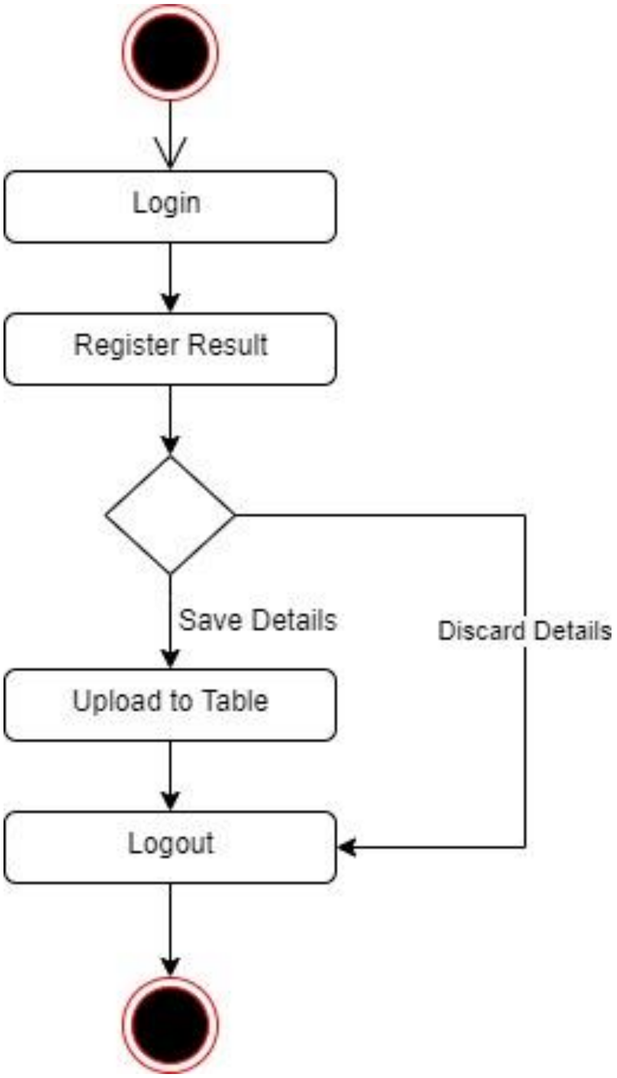


Fig 3.5 Register Result Activity Diagram

3.4 Database Design

Input specification is the logical presentation of how data is stored in the computer's memory. SQL standards are vital so that structured data will be uniform and independent of applications, the flexibility encountered in the use of the system as well as the ease in recalling and reading the data and ensuring applicability through the internet, the input specifications used in this project work are presented below:

i. Accounts Table: contains basic information about all system users (Admin and Recruiters/Organization).

ii. Result Table: contains needed information of every student results.

Table 3.1 Account Table input specification table

FIELD NAME	DATA TYPE	LENGTH	DESCRIPTION
Email	String	150	Email for login (case sensitive)
Password	String	150	Access Code (case sensitive)
Firstname	String	150	Admin user first name
Lastname	String	150	Admin user last name
Organization name	String	150	Organization full name
Phone	String	150	User phone number
Picture	String	-	Organization logo or Admin profile picture
is_staff	Boolean	1	Key to differentiate system users
acct_id	String	64	A unique string for identifying users

Primary key: acct_id

TABLE 3.2 Result Input Specification Table

Field NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
Full name	String	50	Student full name
Programme	String	20	Student enrollment programme
Department	String	100	Student department
Grade	String	20	Student graduate grade
Cert_no	String	-	To describe the tutorial
Result_id	String	1	A unique string for identifying results
Date	Date	-	Date the result was issued

Primary Key: Result_id

3.5 Output Design

This declares and show the result obtained from the input specified. The output product by the automated system depends on the input. Below is the output specification.

Table 3.3 Account Output Design Table

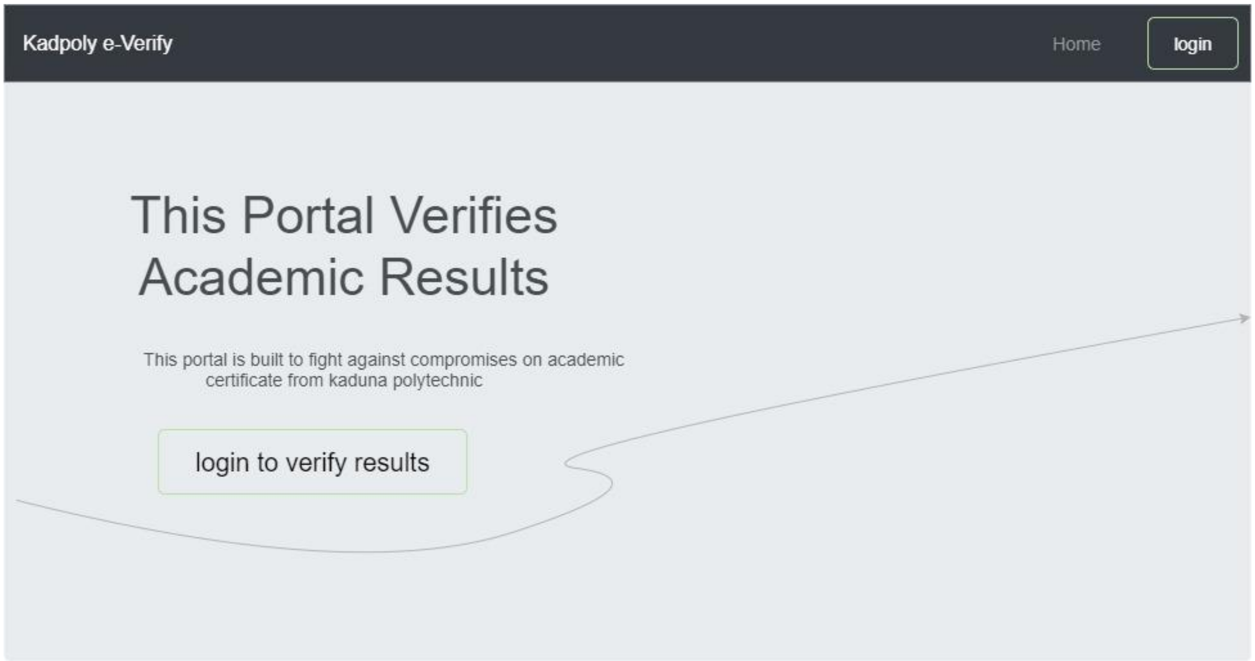
Email	Password	Is_Staff	Acct_id	Firstname	Lastname	Org_name	Phone	Picture
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

Table 3.4 Result Output Design Table

Result_id	Fullname	Programme	Department	Grade	Cert_no	Date
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX

3.6 Input & User Interface Design

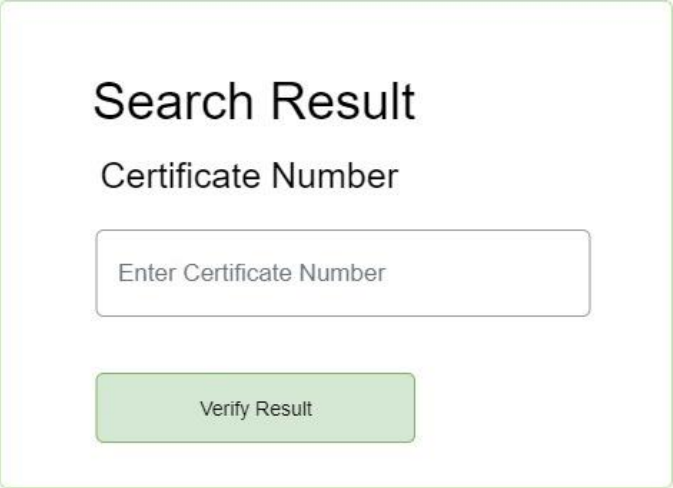
This displays the pictorial representation of the system interface, the interface is going to be designed in a way that it will be user friendly, responsive and attractive. It will also be well secured such that login will be required to access some level of contents. The designs are aided by a mid-fidelity wireframing tool called Draw.io



This Portal Verifies Academic Results

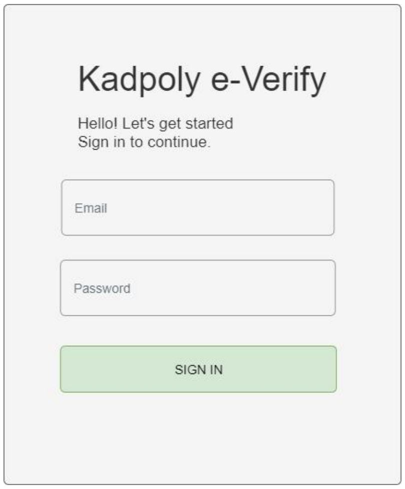
This portal is built to fight against compromises on academic certificate from kaduna polytechnic

Figure 3.6.1 Home page



Search Result

Certificate Number



Kadpoly e-Verify

Hello! Let's get started
Sign in to continue.

Figure 3.6.2 Verify Result Form

Figure 3.6.3 User Login Page

3.7 System Requirement

All software system developed has a predetermined system requirement on which it has been designed to operate on for maximum performance. However, the system requirements are the minimum hardware and software required for the smooth operation of the system that is designed.

3.7.1 Hardware Requirement

System Hardware Requirement;

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

3.7.2 Software Requirement

Software Requirement;

a. At least window 7 OS.
b. Browsers includes: Chrome, Firefox.
c. Python installation
d. Vs Code installation

3.8 Choice of Programming Language

Various types of programming language exist that could have been used in writing this tutorial application but the choices of programming languages used involve HTML5, CSS3, JavaScript, Python (Django) and SQLite. The reasons for choosing these programming languages is that it is a web-based application and require web programming languages.

CHAPTER THREE

METHODOLOGY AND DESIGN

3.1 Introduction

Methodology is a way of careful study or investigation especially in order to discover new fact or information, hence research methodology should be sound enough to make attainment of the set objectives possible with specific components such as methods of data collection and design. This chapter entails the System Modelling (Use case, activity and class diagrams) as well as the input/output specifications and system requirement for the design of this Kadpoly result e-Verify portal.

3.2 Method of Data Collection

In executing any system, one has to have an insight of what is happening, it is important that information and fact about the existing system is gathered. In executing this research, two methods were employed

iii. Observation of the Work Environment.

iv. Documentation

3.2.1 Observation of the Work Environment

This method was employed in gathering information/data for this research by looking at the way in which the manual system was carried out. The glaring issues with the existing system were detected by careful observation Utilizing the observational method can exert varying amounts of control over the environment in which the observation takes place.

3.2.2 Documentation

The Documentation method is a secondary method of data collection. This method involves the use of journals, handbooks, past projects and newspapers. This method of data collection is used because it serves as a basis of reference to existing research work. This includes internet which is a method of data collection the web was used in sourcing for information on areas that seems difficult or confusing in order to achive an alternative, workable result verification system (Kadpoly result e-Verify).

3.3 System Modeling

System model is the conceptual model as a result of the system modelling that describe and represent a system. A system describes a relationship between any set of components to achieve some common objective. The UML applied in this new design include Use Case Diagram, Class Diagram and Activity Diagrams. In this research work, a Unified Modelling Language (UML) is used.

3.3.1 Use Case Diagram

The purpose of a Use Case Diagram is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

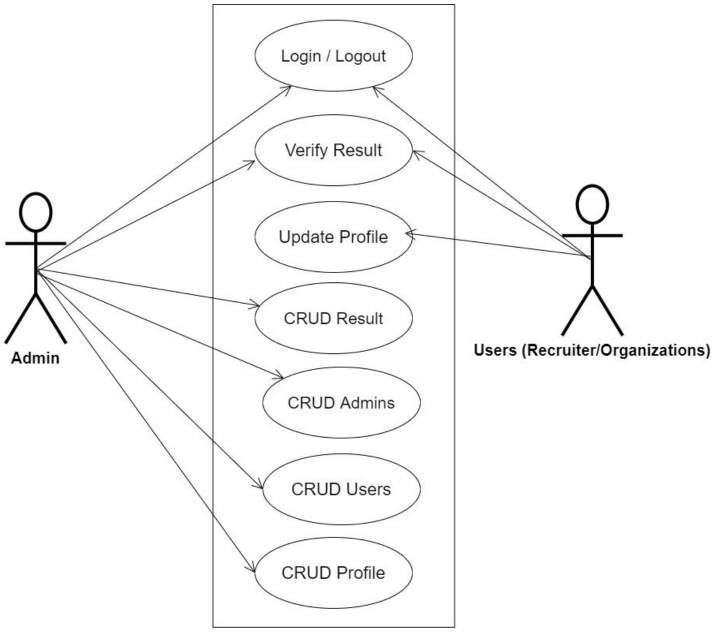


Fig 3.1 System Use Case Diagram

3.3.2 Class Diagram

Class diagram in the Unified Modeling Language (UML) is an implementation of independent view of how the system interface will be, each class with its own attributes and how they are related to each other. Class diagrams are visual representations of the static structure and composition of a particular system using the conventions set by the Unified Modeling Language (UML).

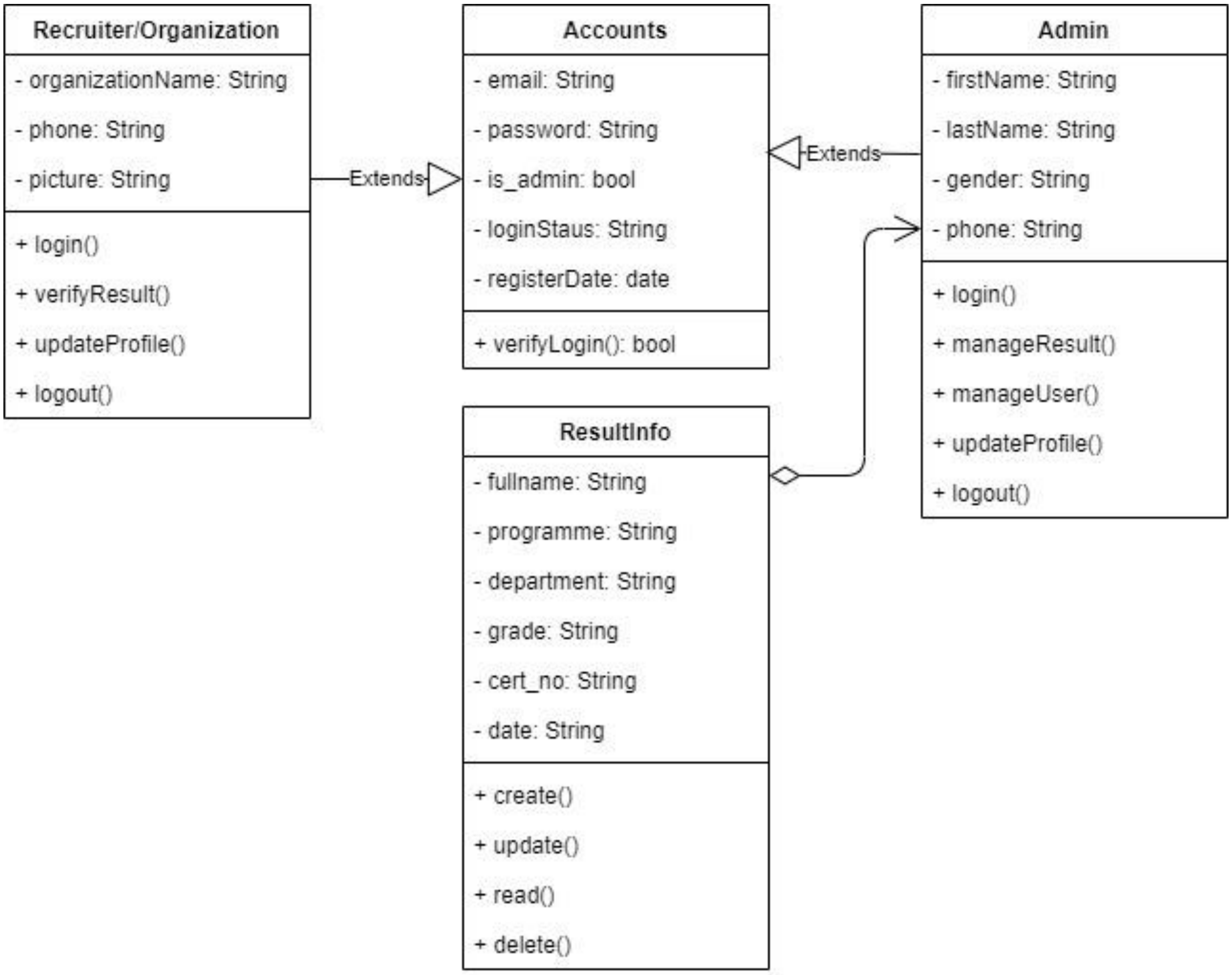


Fig 3.2 System Class Diagram

3.3.3 Activity Diagram

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 username and password must be accurate to gain access.

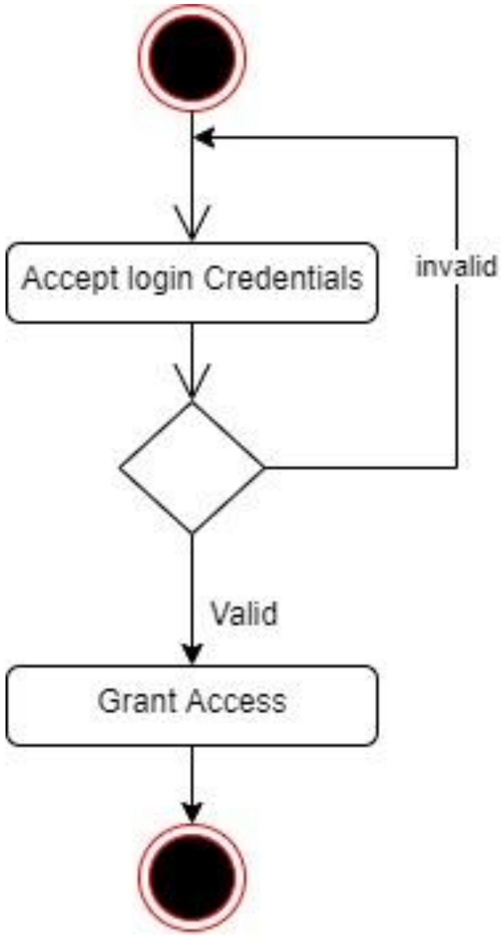


Fig 3.3 Login Activity Diagram

Verify Result

The process for verifying result is depicted in the diagram below; the user must be authenticated to verify result.

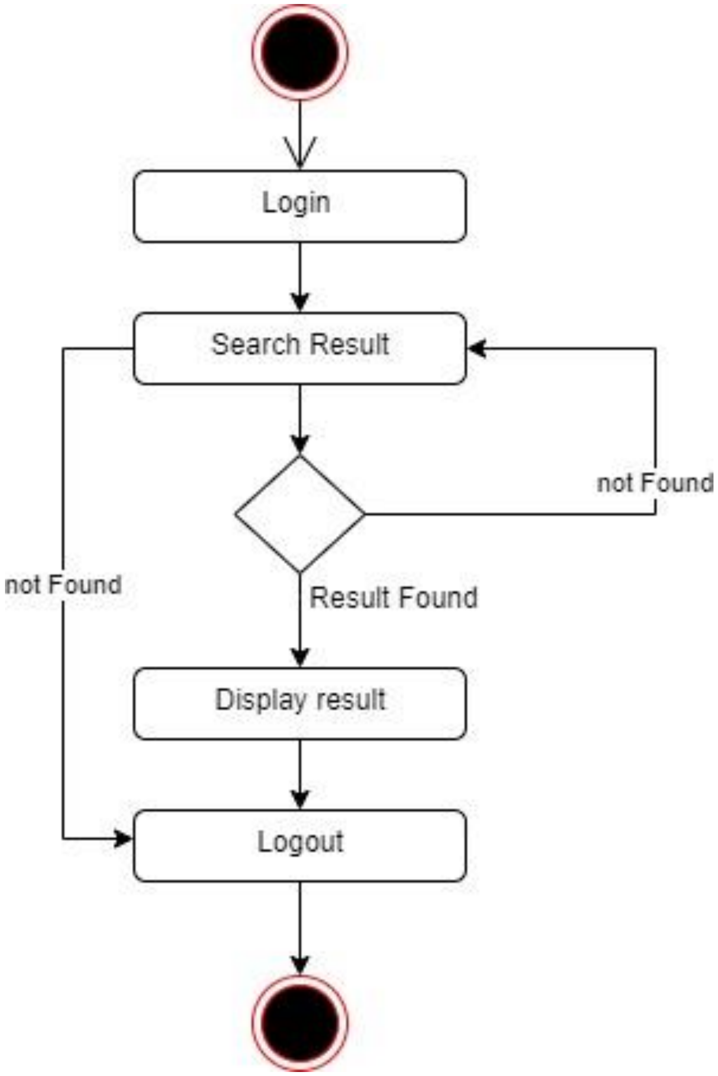


Fig 3.4 Verify Result Activity Diagram

Register Result

The process for registering result is depicted in the diagram below; the user must be authenticated and authorized to perform the registration of result.

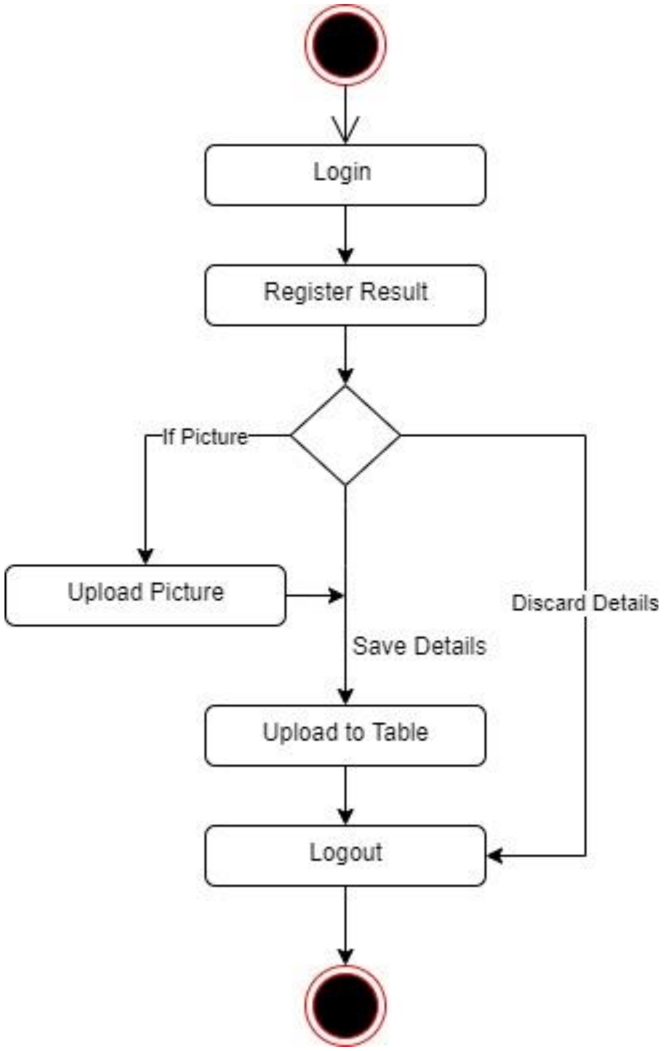


Fig 3.5 Register Result Activity Diagram

3.4 Database Design

Input specification is the logical presentation of how data is stored in the computer's memory. SQL standards are vital so that structured data will be uniform and independent of applications, the flexibility encountered in the use of the system as well as the ease in recalling and reading the data and ensuring applicability through the internet, the input specifications used in this project work are presented below:

iii. Accounts Table: contains basic information about all system users (Admin and Recruiters/Organization).

iv. Result Table: contains needed information of every student results.

Table 3.1 Account Table input specification table

FIELD NAME	DATA TYPE	LENGTH	DESCRIPTION
Email	String	150	Email for login (case sensitive)
Password	String	150	Access Code (case sensitive)
Firstname	String	150	Admin user first name
Lastname	String	150	Admin user last name
Organization name	String	150	Organization full name
Phone	String	150	User phone number
Picture	String	-	Organization logo or Admin profile picture
is_staff	Boolean	1	Key to differentiate system users
acct_id	String	64	A unique string for identifying users

Primary key: acct_id

TABLE 3.2 Result Input Specification Table

Field NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
Full name	String	50	Student full name
Date of Birth	Date	100	Student date of birth
Programme	String	20	Student enrollment programme
Department	String	100	Student department
Grade	String	20	Student graduate grade
Cert_no	String	-	To describe the tutorial
Result_id	String	1	A unique string for identifying results
Date	Date	-	Date the result was issued

Primary Key: Result_id

3.5 Output Design

This declares and show the result obtained from the input specified. The output product by the automated system depends on the input. Below is the output specification.

Table 3.3 Account Output Design Table

Email	Password	Is_Staff	Acct_id	Firstname	Lastname	Org_name	Phone	Picture
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	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

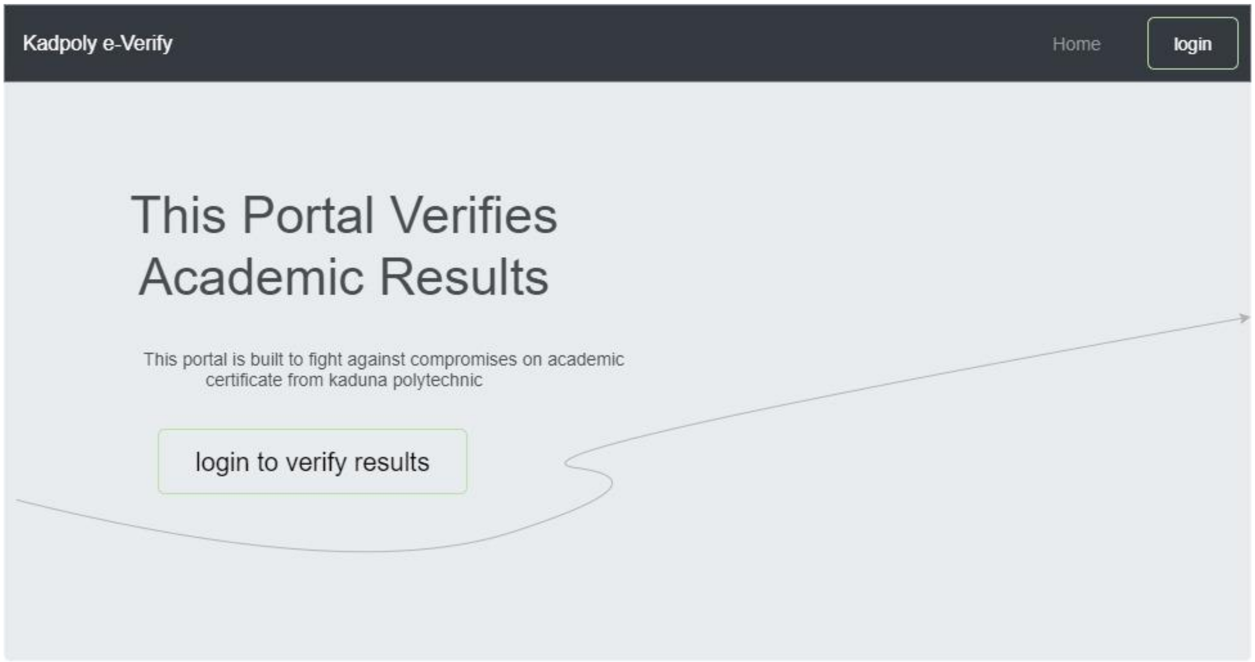
Table 3.4 Result Output Design Table

Result_id	Fullname	Programme	Department	Grade	Cert_no	Date
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX
XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX

XXXXX	XXXXX	XXXXXX	XXXXX	XXXXX	XXXXX	XXXXX

3.6 Input & User Interface Design

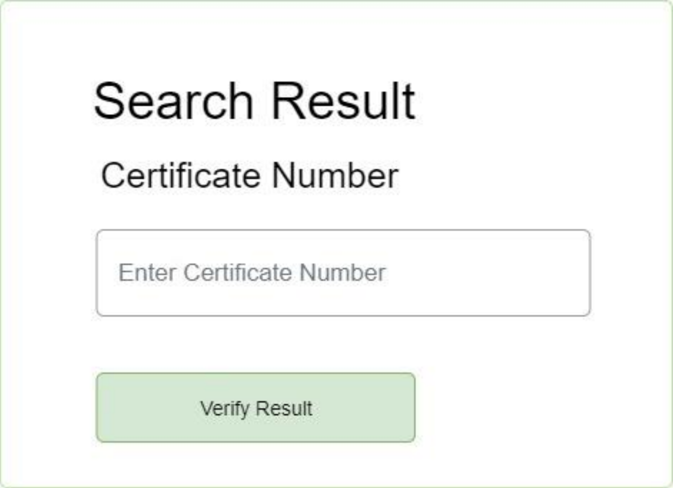
This displays the pictorial representation of the system interface, the interface is going to be designed in a way that it will be user friendly, responsive and attractive. It will also be well secured such that login will be required to access some level of contents. The designs are aided by a mid-fidelity wireframing tool called Draw.io



This Portal Verifies Academic Results

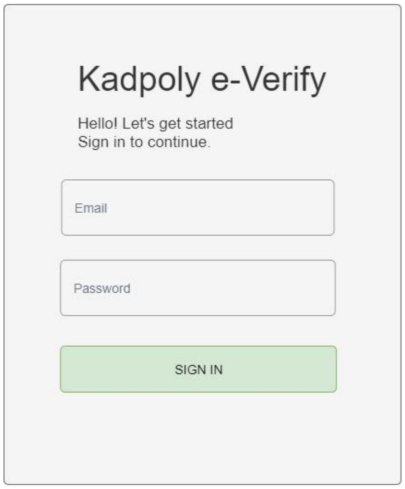
This portal is built to fight against compromises on academic certificate from kaduna polytechnic

Figure 3.6.1 Home page



Search Result

Certificate Number



Kadpoly e-Verify

Hello! Let's get started
Sign in to continue.

Figure 3.6.2 Verify Result Form **Figure 3.6.3 User Login Page**

3.7 System Requirement

All software system developed has a predetermined system requirement on which it has been designed to operate on for maximum performance. However, the system requirements are the minimum hardware and software required for the smooth operation of the system that is designed.

3.7.1 Hardware Requirement

System Hardware Requirement;

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

3.7.2 Software Requirement

Software Requirement;

e. At least window 7 OS.
f. Browsers includes: Chrome, Firefox.
g. Python installation
h. Vs Code installation

3.8 Choice of Programming Language

Various types of programming language exist that could have been used in writing this tutorial application but the choices of programming languages used involve HTML5, CSS3, JavaScript, Python (Django) and SQLite. The reasons for choosing these programming languages is that it is a web-based application and require web programming languages.

CHAPTER FOUR

SYSTEM IMPLEMENTATION EVALUATION

4.1 Introduction

This section describes in concise detail how the new system is implemented for effective operation. It shows samples of the working (new) system designed and how the system is to be installed.

4.2 System Testing and Evaluation

There are many reasons to conduct the testing for the developed system because is only through testing that we can be able to analyze any problem in the new system and provide solutions to these problems This project employed both unit and integration testing to ensure the effectiveness and efficiency of design and to ensure that the new system meets its required functionalities and is error-free.

Unit Testing

In this section, individual units or single components of the system are tested independently to ensure that individual phases are working effectively without errors.

Integration Testing

Testing of the program was implemented using integration testing all the units were put together as one so they function as one. The link between the various units was tested to be sure that they are correctly integrated, and also to be sure that the units can function correctly together as one.

4.3 System Conversion Plan

The conversion of the new system is a parallel approach i.e. both the existing and new system will be used concurrently before it finally has a good stand in the system. This is used because having jumped out to a new system directly without the existing system can cause a breakdown to every record already taken during the cause of using the new system alone if there is any problem in the program maintenance.

4.4 System Installation

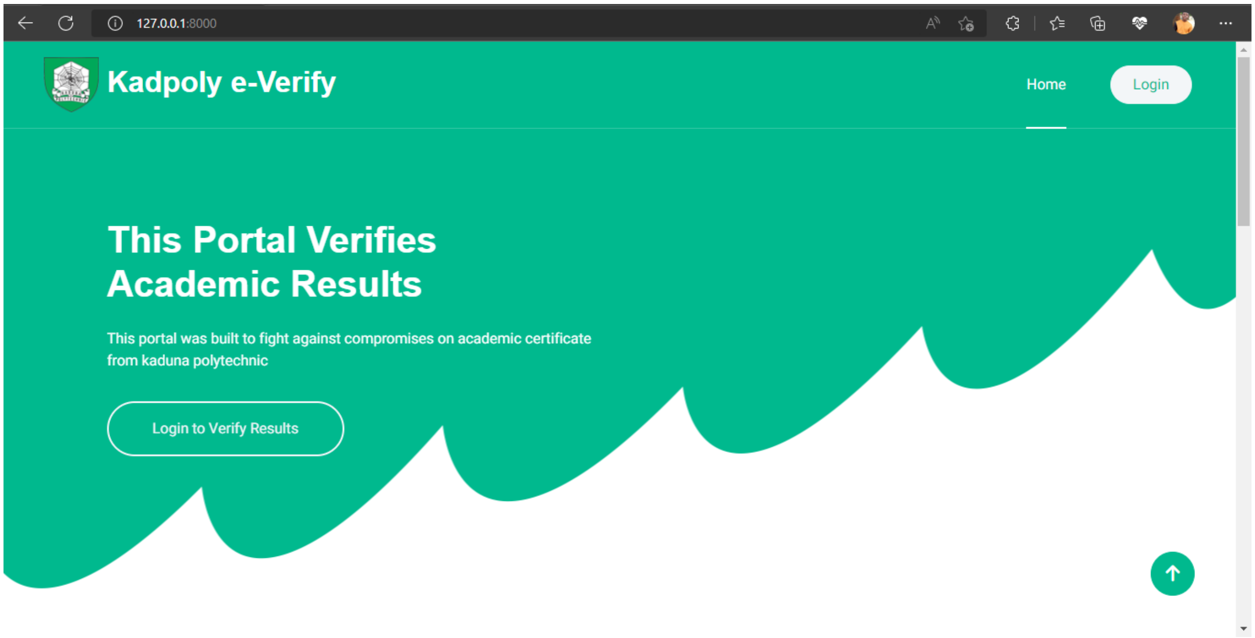
In order to use the proposed application on any computer system, the following steps need to be taken:

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 the 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.

4.5 Security Measures

Since the scope of the website is public, some of the information such as the index page, login page e.t.c are available to anyone who visits the website. But some other information and functionalities are restricted to some and not all who visit the website. The restrictions are carried out by the use of passwords which gives different levels of access to users. The admin holds the highest level of access, followed by the organizations with lesser access.

4.6 Sample Outputs

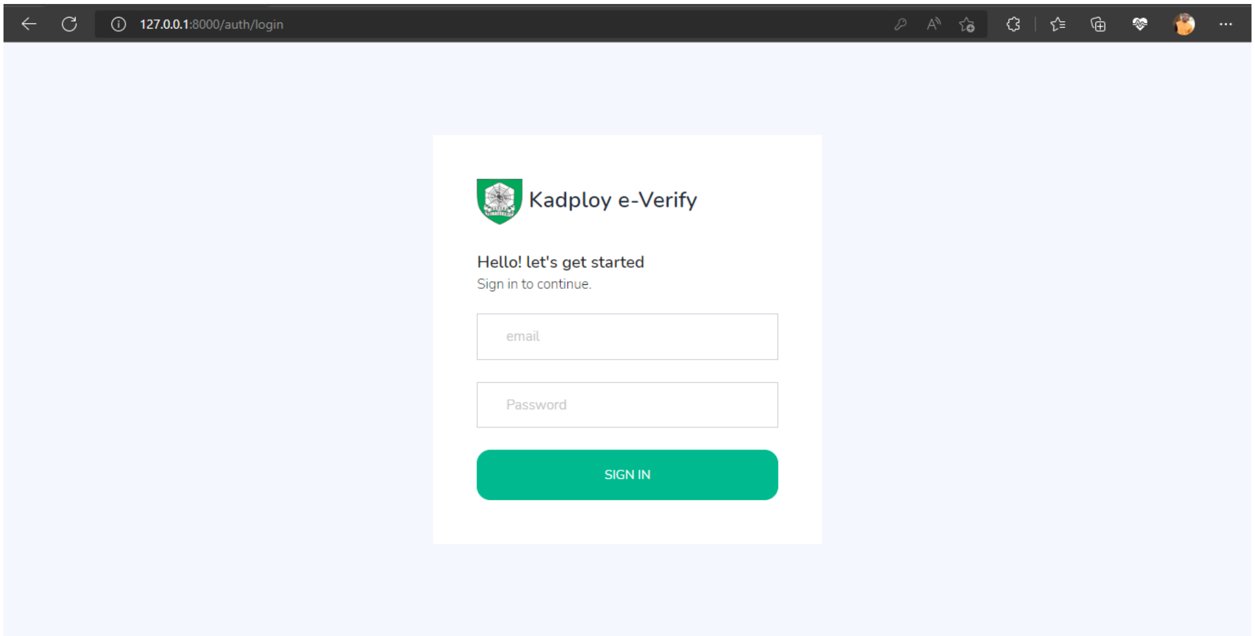


Kadpoly e-Verify

This Portal Verifies Academic Results

This portal was built to fight against compromises on academic certificate from kaduna polytechnic

Fig 4.1 Home Page

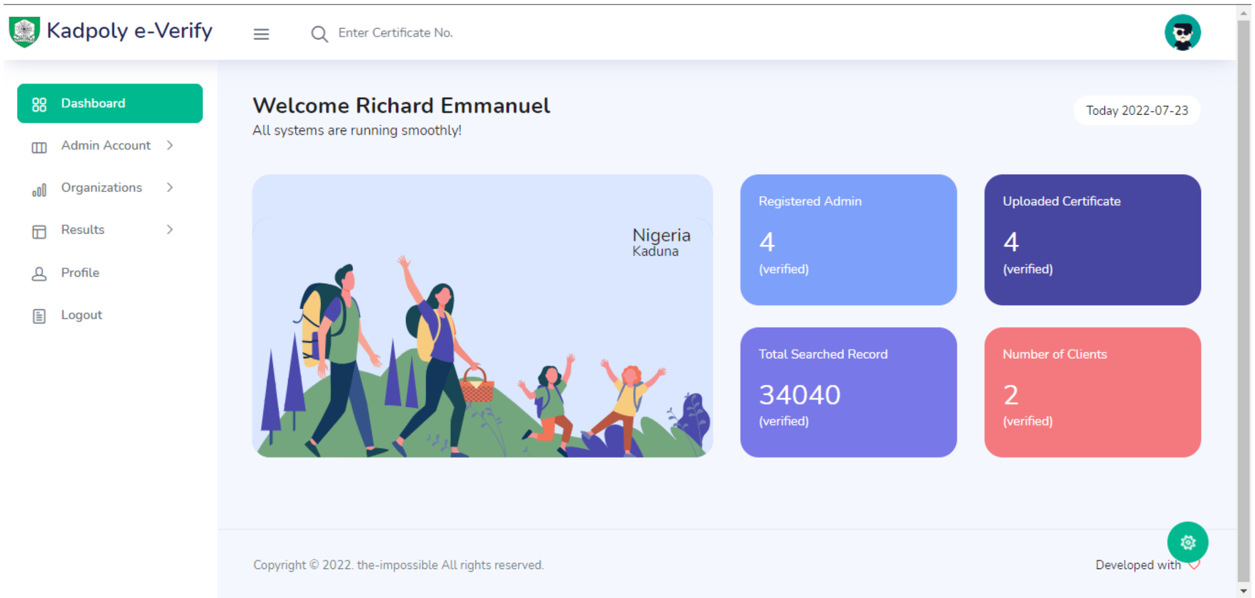


Kadploy e-Verify

Hello! let's get started

Sign in to continue.

Fig 4.2 Login Form



Welcome Richard Emmanuel

All systems are running smoothly!

Today 2022-07-23

Registered Admin 4 (verified)

Uploaded Certificate 4 (verified)

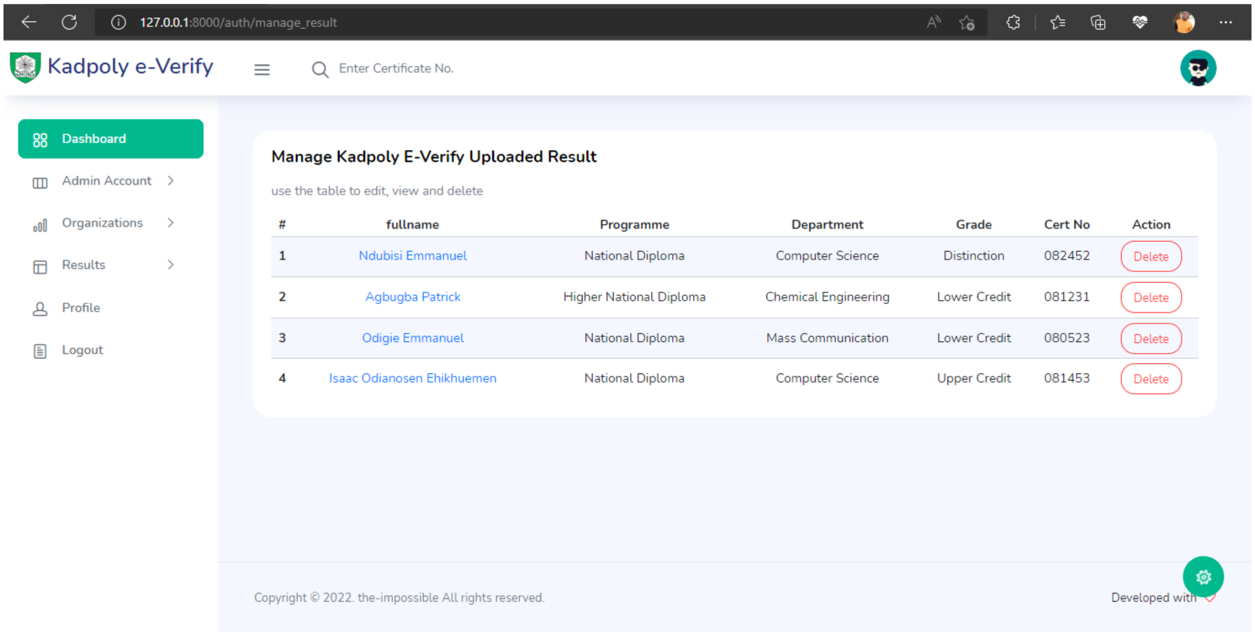
Total Searched Record 34040 (verified)

Number of Clients 2 (verified)

Copyright © 2022. the-impossible All rights reserved.

Developed with ♡

Fig 4.3 Admin Dashboard



Manage Kadpoly E-Verify Uploaded Result

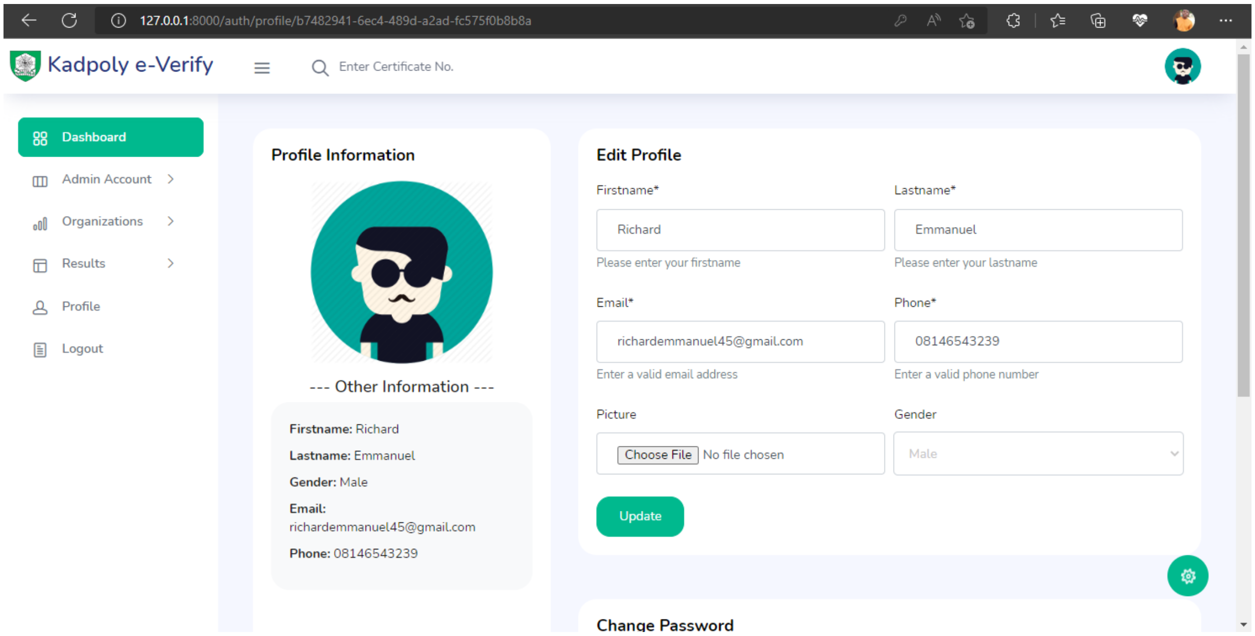
use the table to edit, view and delete

#	fullname	Programme	Department	Grade	Cert No	Action
1	Ndubisi Emmanuel	National Diploma	Computer Science	Distinction	082452	Delete
2	Agbugba Patrick	Higher National Diploma	Chemical Engineering	Lower Credit	081231	Delete
3	Odigie Emmanuel	National Diploma	Mass Communication	Lower Credit	080523	Delete
4	Isaac Odianosen Ehikhuemen	National Diploma	Computer Science	Upper Credit	081453	Delete

Copyright © 2022. the-impossible All rights reserved.

Developed with ♡

Fig 4.4 Manage Result



Profile Information

--- Other Information ---

Firstname: Richard

Lastname: Emmanuel

Gender: Male

Email: richardemmanuel45@gmail.com

Phone: 08146543239

Edit Profile

Firstname*: Richard
Please enter your firstname

Lastname*: Emmanuel
Please enter your lastname

Email*: richardemmanuel45@gmail.com
Enter a valid email address

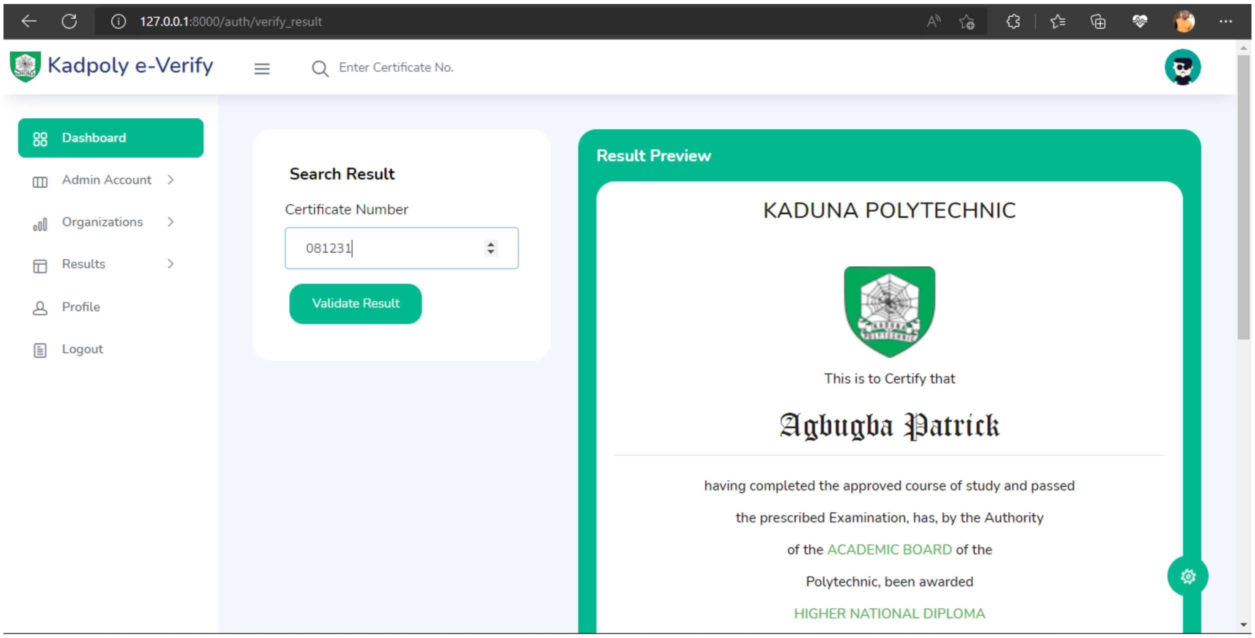
Phone*: 08146543239
Enter a valid phone number

Picture: Choose File No file chosen

Gender: Male

Change Password

Fig 4.5 Profile Page



Search Result

Certificate Number: 081231

Result Preview

KADUNA POLYTECHNIC

This is to Certify that

Agbugba Patrick

having completed the approved course of study and passed the prescribed Examination, has, by the Authority of the ACADEMIC BOARD of the Polytechnic, been awarded

HIGHER NATIONAL DIPLOMA

Fig 4.6 Result Verification Page

CHAPTER FIVE

SUMMARY CONCLUSION AND RECOMMENDATION

5.1 Summary

This project is titled "Result Validation and Verification System for Kaduna Polytechnic (Kadpoly e-Verify)" aimed to efficiently validate results from Kaduna Polytechnic, thereby reducing the cost of parsing information and establishing a non-location-based platform for organizations, employers, and administrators from Kaduna polytechnic to help fight against forgery of results. This work examines the challenges of the existing system and provides suitable solutions.

5.2 Conclusion

This research work is an effort toward the elimination of fake certificates in learning institutions. As discussed in the introduction chapter of this study, verification of academic certificates is one of the important research areas today. This work contributes towards solving problems in academic fraud. A part of this work focused on the application of the proposed prototype as proof of concept. In our proof of concept, the prototype was able to verify the student's academic details as they are in the database which was uploaded by the school.

5.2 Recommendation

Based on my experience during the solving problem, I would like to recommend that more research be done in better data verification processes and a lot of literature review done for related work. After successfully implementing and testing the project prototype, in order to realize the intended purpose of the study I recommend the prototype to be adopted by other Higher learning institutions and secondary education institutions as well Organizations and recruiting agencies are the most targeted users of the system.

References

Clement, M. (2018). Academic Certificate Record Verifying Platform: department of computing science studies (CSS) of faculty of science and technology

Dinesh Kumar, K., Senthil, P., & Manoj Kumar, D. S. (2020). Educational certificate verification system using blockchain. *International Journal of Scientific and Technology Research, 9*(3), 82–85.

Emele, I. C., Oguoma, S. I., Uka, K. K., & Nwaoha, E. C. (2020). An Enhanced Web Base Certificate Verification System. *OALib, 07*(07), 1–15.

Oblikwu, P., & Dekera, K. (2019). A Generic Certificate Verification System for Nigerian Universities. *International Journal of Computer Science and Mobile Computing, 8*(2019), 137–148.

SEBoK v. 1.9 , November 17 2017, System verification, Retrieved from http://sebokwiki.org/wiki/Systems_Engineering_and_Management on 26th May 2018

Thua Huynh, T., & Khoa Pham, D. (2019). EUNICERT: Ethereum Based Digital Certificate Verification System. International Journal of Network Security & Its Applications

Appendix

view.py

```
# My Django imports
from django.shortcuts import render, redirect, reverse
from django.views import View
from django.contrib.auth import authenticate, login, logout
from datetime import date
from django.contrib import messages
from django.contrib.messages.views import SuccessMessageMixin
from django.core.exceptions import ObjectDoesNotExist
from django.views.generic import ListView, DetailView, CreateView, UpdateView,
DeleteView
from django.http import HttpResponse
from django.contrib.auth.mixins import LoginRequiredMixin

# My App imports
from e_verify_auth.models import Accounts
from e_verify_app.models import ResultInformation
from e_verify_auth.forms import AccountCreationForm, OrganizationForm,
AccountUpdateForm, OrganizationUpdateForm, ResultForm, EditResultForm

# Create your views here.
class DashboardView(LoginRequiredMixin, View):
    login_url = 'auth:login'
    def get(self, request):
        context = {
            'time':date.today().strftime("%Y-%m-%d"),
            'admin':Accounts.objects.all().count(),
            'clients':Accounts.objects.filter(is_staff=False).count(),
            'cert': ResultInformation.objects.all().count(),
        }
        return render(request, 'auth/dashboard.html', context)

class LoginView(View):
    def get(self, request):
        return render(request, 'auth/login.html')

    def post(self, request):
        email = request.POST.get('email').strip().lower()
        password = request.POST.get('password').strip()


        if email and password:
            # Authenticate user
            user = authenticate(request, email=email, password=password)

            if user:
                if user.is_active:
                    login(request, user)
                    messages.success(request, f'You are now signed in {user}')
                    return redirect('auth:dashboard')
                else:
                    messages.warning(request, 'Account not active contact the
administrator')
                    return redirect('auth:login')
            else:
                messages.warning(request, 'Invalid login credentials')
                return redirect('auth:login')
        else:
            messages.error(request, 'All fields are required!!')
            return redirect('auth:login')

class CreateAdminView(LoginRequiredMixin, SuccessMessageMixin, CreateView):
    login_url = 'auth:login'
    model = Accounts
    form_class = AccountCreationForm
    template_name = 'auth/create_admin.html'
    success_message = "Account created successfully!"

    def get_success_url(self):
        return reverse("auth:manage_admin")

    def form_valid(self, form):
        form.instance.set_password(form.instance.password)
        form.instance.email = form.instance.email.strip().lower()
        form.instance.is_staff = True
        return super().form_valid(form)

class ManageAdminView(LoginRequiredMixin, ListView):
    login_url = 'auth:login'
    model = Accounts
    template_name = "auth/manage_admin.html"

    def get_queryset(self):
        return Accounts.objects.filter(is_staff=True).order_by('-date_joined')

class LogoutView(LoginRequiredMixin, View):
    login_url = 'auth:login'
    def post(self, request):
        logout(request)
        messages.success(request, 'You are successfully logout, to continue login
again')

        return redirect('auth:login')

class RegisterView(SuccessMessageMixin, CreateView):
    model = Accounts
    form_class = OrganizationForm
    template_name = 'auth/register.html'
    success_message = "Account created successfully! You can now login!"

    def get_success_url(self):
        return reverse("auth:login")

    def form_valid(self, form):
        form.instance.set_password(form.instance.password)
        form.instance.email = form.instance.email.strip().lower()
        return super().form_valid(form)

class CreateOrgView(LoginRequiredMixin, RegisterView):
    login_url = 'auth:login'
    template_name = 'auth/create_org.html'

    def get_success_url(self):
        return reverse("auth:manage_org")

class ManageOrgUserView(ManageAdminView):
    login_url = 'auth:login'
    template_name = "auth/manage_org.html"

    def get_queryset(self):
        return Accounts.objects.filter(is_staff=False).order_by('-date_joined')

class DeleteUserView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):
    login_url = 'auth:login'
    model = Accounts
    success_message = "Account deleted successfully!"

    def get_success_url(self):
        return reverse("auth:manage_admin")

class DeleteOrgView(DeleteUserView):
    login_url = 'auth:login'
    def get_success_url(self):
        return reverse("auth:manage_org")

class ProfileView(LoginRequiredMixin, SuccessMessageMixin, View):
    login_url = 'auth:login'
    def get(self, request, pk):
        try:
            user = Accounts.objects.get(pk=pk)
            if user.is_staff:
                context = {
                    'form': AccountUpdateForm(instance=user),
                    'user': user,
                }
            else:
                context = {
                    'form': OrganizationUpdateForm(instance=user),
                    'user': user,
                }
            return render(request, 'auth/profile.html', context)
        except ObjectDoesNotExist:
            messages.error(request, 'User account not found!')
            return redirect('auth:dashboard')

    def post(self, request, pk):
        try:
            user = Accounts.objects.get(pk=pk)
            if 'password' in request.POST:
                password1 = request.POST.get('password1')
                password2 = request.POST.get('password2')

                if user.is_staff:
                    context = {
                    'form': AccountUpdateForm(instance=user),
                    'user': user,
                }
                else:
                    context = {
                        'form': OrganizationUpdateForm(instance=user),
                        'user': user,
                    }

                if password1 and password2:
                    if password1 != password2:
                        messages.error(request, 'Passwords does not match!')
                        return redirect('auth:profile', pk)
                    if len(password1) < 6 :
                        messages.error(request, 'Password too short, ensure at
least 6 characters!')
                        return redirect('auth:profile', pk)

                    user.set_password(password1)
                    user.save()

                    messages.success(request, 'Password reset successful!!')
                    if request.user == user:
                        return redirect('auth:login')

                    if request.user.is_superuser:
                        return redirect('auth:profile', pk)
                    return redirect('auth:login')
            else:
                if user.is_staff:
                    form = AccountUpdateForm(request.POST, request.FILES,
instance=user)
                else:
                    form = OrganizationUpdateForm(request.POST, request.FILES,
instance=user)

                if form.is_valid() :
                    form.save()
                    messages.success(request, 'Profile updated successfully!')
                    return redirect('auth:profile', pk)

                messages.error(request, 'your response contains invalid data!')
                return render(request, 'auth/profile.html', {'form':form,
'user':user})
        except ObjectDoesNotExist:
            messages.error(request, 'User account not found!')
            return redirect('auth:dashboard')

class UploadResultView(LoginRequiredMixin, SuccessMessageMixin, CreateView):
    login_url = 'auth:login'
    model = ResultInformation
    form_class = ResultForm
    template_name = 'auth/upload_result.html'
    success_message = "Result Information uploaded!"

    def get_success_url(self):
        return reverse("auth:manage_result")
```

```
class ManageResultView(LoginRequiredMixin, View):
    login_url = 'auth:login'
    def get(self, request):
        return render(request, 'auth/manage_result.html')

class ListResultView(ManageAdminView):
    login_url = 'auth"login'
    template_name = "partials/result_list.html"

    def get_queryset(self):
        return ResultInformation.objects.all().order_by('-date')

class ResultEditForm(LoginRequiredMixin, View):
    login_url = 'auth:login'
    def get(self, request, pk):
        result = ResultInformation.objects.get(pk=pk)
        form = EditResultForm(instance=result)
        return render(request, 'auth/result_form.html', {'form':form,
'result':result})

    def post(self, request, pk):
        result = ResultInformation.objects.get(pk=pk)
        form = EditResultForm(request.POST, instance=result)
        if form.is_valid():
            form.save()
            messages.success(request, 'Result has been updated!')
            return HttpResponse(status=204, headers={'Hx-Trigger':'listChanged'})
        messages.error(request, f'{form.errors.as_text()}')
        return HttpResponse(status=204, headers={'Hx-Trigger':'listChanged'})

class DeleteResultView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):
    login_url = 'auth:login'
    model = ResultInformation
    success_message = "Result deleted successfully!"

    def get_success_url(self):
        return reverse("auth:manage_result")

class VerifyResult(LoginRequiredMixin, View):
    login_url = 'auth:login'
    def get(self, request):
        return render(request, 'auth/verify_result.html')

    def post(self, request):
        qs =  request.POST.get('search')
        result = ResultInformation.objects.filter(cert_no=qs)

        if result:
            return render(request, 'partials/result_content.html',
context={'result':result[0], 'qs':qs})
        else:
            messages.error(request, 'Result not found! try inputting a valid
cert_no')
        return render(request, 'partials/result_empty.html', {'qs':qs})

class SearchResult(LoginRequiredMixin, View):
    login_url = 'auth:login'
    def post(self, request):
        qs =  request.POST.get('qs')
        result = ResultInformation.objects.filter(cert_no=qs)
        if result:
            return render(request, 'auth/verify_result.html',
context={'result':result[0], 'qs':qs})
        else:
            messages.error(request, 'Result not found! try inputting a valid
cert_no')
        return render(request, 'auth/verify_result.html', {'qs':qs})
```

Home Page

```
{% extends "base.html" %}
{% load static %}
{% block title %} Home{% endblock %}
{% block head %}
    {% include "partials/head.html" %}
{% endblock %}
{% block body %}
    <div class="container-xxl bg-white p-0">
        <!-- Spinner Start -->
        <div id="spinner" class="show bg-white position-fixed translate-middle w-
100 vh-100 top-50 start-50 d-flex align-items-center justify-content-center">
            <div class="spinner-border text-primary" style="width: 3rem; height:
3rem;" role="status">
                <span class="sr-only">Loading...</span>
            </div>
        </div>
        <!-- Spinner End -->

        {% include "partials/nav.html" %}
        <!-- About Start -->
        <div class="container-xxl">
            <div class="container">
                <div class="row g-5 align-items-center">
                    <div class="col-lg-6 wow zoomIn" data-wow-delay="0.1s">
                        <img class="img-fluid" src="{% static 
'img/certificate.png' %}">
                    </div>
                    <div class="col-lg-6 wow fadeInUp" data-wow-delay="0.1s">
                        <div class="d-inline-block border rounded-pill text-
primary px-4 mb-3">About KASU e-Verify</div>
                        <h2 class="mb-4"> Project</h2>
                        <p class="mb-4">Lorem ipsum dolor sit amet consectetur 
adipisicing elit. Hic minima optio asperiores ipsam repudiandae .:</p>
                        <div class="row g-3 mb-4">
                            <div class="col-12 d-flex">
                                <div class="flex-shrink-0 btn-lg-square rounded-
circle bg-primary">
                                    <i class="fa fa-user-tie text-white"></i>
                                </div>
                                <div class="ms-4">
                                    <h6>vel voluptates</h6>
                                    <!-- <span>CST20ND0558</span> -->
                                </div>
                            </div>
                            <div class="col-12 d-flex">
                                <div class="flex-shrink-0 btn-lg-square rounded-
circle bg-primary">
                                    <i class="fa fa-user-tie text-white"></i>
                                </div>
                                <div class="ms-4">
                                    <h6>um aliquid eaque obcaecati</h6>
                                    <!-- <span>CST20HND0558</span> -->
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <!-- About End -->

        <!-- Service Start -->
        <div class="container-xxl py-6">
```

```
            <div class="container">
                <div class="mx-auto text-center wow fadeInUp" data-wow-
delay="0.1s" style="max-width: 600px;">
                    <div class="d-inline-block border rounded-pill text-primary
px-4 mb-3">Features of KASU e-Verify</div>
                    <h2 class="mb-5">KASU e-Verify possess some features that
makes it flexible</h2>
                </div>
                <div class="row g-4">
                    <div class="col-lg-4 col-md-6 wow fadeInUp" data-wow-
delay="0.1s">
                        <div class="service-item rounded h-100">
                            <div class="d-flex justify-content-between">
                                <div class="service-icon">
                                    <i class="fa fa-user-tie fa-2x"></i>
                                </div>
                                <a class="service-btn">
                                    <i class="fa fa-link fa-2x"></i>
                                </a>
                            </div>
                            <div class="p-5">
                                <h5 class="mb-3">Nice User Interface</h5>
                                <span>KASU e-Verify provides a nice user friendly
and easy to use user interface.</span>
                            </div>
                        </div>
                    </div>
                    <div class="col-lg-4 col-md-6 wow fadeInUp" data-wow-
delay="0.3s">
                        <div class="service-item rounded h-100">
                            <div class="d-flex justify-content-between">
                                <div class="service-icon">
                                    <i class="fa fa-chart-pie fa-2x"></i>
                                </div>
                                <a class="service-btn">
                                    <i class="fa fa-link fa-2x"></i>
                                </a>
                            </div>
                            <div class="p-5">
                                <h5 class="mb-3">Dynamic Search</h5>
                                <span>Fast and no reload on the search
page.</span>
                            </div>
                        </div>
                    </div>
                    <div class="col-lg-4 col-md-6 wow fadeInUp" data-wow-
delay="0.6s">
                        <div class="service-item rounded h-100">
                            <div class="d-flex justify-content-between">
                                <div class="service-icon">
                                    <i class="fa fa-chart-line fa-2x"></i>
                                </div>
                                <a class="service-btn">
                                    <i class="fa fa-link fa-2x"></i>
                                </a>
                            </div>
                            <div class="p-5">
                                <h5 class="mb-3">Security</h5>
                                <span>KASU e-Verify is a secured
application.</span>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <!-- Service End -->

       {% include "partials/footer.html" %}
    </div>
    {% include "partials/script.html" %}
{% endblock %}
```