COMBATING CERTIFICATE FRAUD IN ACADEMIC INSTITUTION: A CLOUD BASED VERIFICATION SYSTEM

BY

UGBERO DAVID EGUONO

(PSC1506330)

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCES,

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

JULY, 2021

COMBATING CERTIFICATE FRAUD IN ACADEMIC INSTITUTION: A CLOUD BASED VERIFICATION SYSTEM

BY

UGBERO DAVID EGUONO

(PSC1506330)

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN CITY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE.

JULY, 2021

CERTIFICATION

This is to certify that this project work has been carried out by UGBERO DAVID EGUONO with Matriculation Number PSC1506330 under my supervision, and it is adequate and satisfactory both in scope and content, for the award of Bachelor of Science (B.sc) Degree in Computer Science of the University of Benin.

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.sc) Degree in Computer Science of the University of Benin.

Prof. F. I. AMADIN **DATE**

Head of Department

DEDICATION

I dedicate this work to God Almighty for giving me the needed strength to carry out the work and for his care, protection throughout my stay in the prestigious University of Benin.

I also dedicate this work to my parents who successfully made me a graduate today and to my siblings too.

ACKNOWLEDGEMENT

My utmost appreciation goes to God Almighty who has seen me through all my education years. Praised be his name forever, Amen.

Words are not enough to express my profound gratitude to my project supervisor, F.O Oliha, Ph.D. for his patience, motivation and immense support all through my stay in the University of Benin.

Special gratitude goes to the Head of Department of Computer Science, Prof. Dr, F.I Amadin, my project coordinator, and all the lecturers in the Department of Computer Science, University of Benin, for their support and inspiration in one way or the other: Prof E.A Onibere, Prof. (Mrs) S.C Chiemeke, Prof (Mrs) V.V.N Akwukwuma, Prof. (Mrs) F.A. Egbokhare, Prof A.A Imiavian, Prof G.O. Ekuobase, Prof. (Mrs) A.O Egwali, Dr. K.C Ukaoha, Dr. (Mrs) S. Konyeha, Dr. E.P Ebietomere, Mr. E.E Obasohan, Dr. (Mrs) R.O Osaseri, Mr, P.E.B Imiefoh, Mr. N.E.O. Agbonlahor, Mr. S.O.P. Oliomogbem Mr. E. Nwelih, Mrs. R.A. Usiobaifo, Mr. E.C. Igodan, Mr. K.O. Otokiti, Miss O.L. Usiosefe, Mr. F. Osagie, Mr. E. Obayagbona, Mrs. T. Agenmonmen, Mrs. O. Aziken, Mrs. Odetayo. God bless you all.

All thanks to my ever loving parents, Mr. and Mrs. James Ugbero for their profound help and support during the course of this project work and my stay in school. All thanks to my siblings: Evelyn, Ufuoma, Blessing, Emmanuel, Peter, Loveth Ugbero for their love, care, understanding and prayers and also to my friends: Peter Avamale, Prosper, Peace, Victory and lots more for their unconditional support throughout my stay in this university. God in his infinite mercy, bless you all.

TABLE OF CONTENT

ABSTRACT

Academic Document fraud in particular has eaten deep into our academic institutions and this increase knows no bound due dearth of systems for proper verification. This research developed a prototype system for combating academic document fraud in tertiary institutions. It made use of Service oriented architecture Modelling Language (SoaML) to model the system. However, it integrates distributed document repository with the interoperable capabilities of web services. The system was designed, implemented, tested and documented. It however possesses the functional capabilities of combating academic certificate fraud bedeviling academic institution across the nation.

CHAPTER ONE

INTRODUCTION

1.1 Background Study

The impact of digital literacy is almost felt across the globe in every economy sectors. This literacy brings about the knowledge of computing technologies and how they can be applied to improve and enhance the computational needs of individuals, communities, industries, organizations and many other firms. Application of computing technologies to our business environment encouraged the development of sophisticated software to help organizations carryout their business transactions and services. Such applications being adopted by many organizations have helped to make their jobs easier and faster. In the banking sector, they make use of an Automated Teller Machine. In the Military, software applications are built to control drones. In the accounting sector, there are applications that help in auditing and other accounting activities, lawyers use history assistance system to help gather relevant cases. In the medical profession, they make use of some sophisticated applications to diagnose patients. In the educational sector there exists many application software to lecture and to manage students' information.

Most tertiary institutions now make use of portals that enable students make payment, register their courses, apply for matriculation and graduation gown, check and print students' result, manage students' records etc. These documents and records are stored on the institution's web portal. However, on the completion of a particular program – undergraduate or postgraduate studies, the institution issues the student a certificate which is a document that serves as a proof that an individual has undertook and completed his study.

There are different kinds of documents issued by diverse organizations, some of which are Birth Certificates, National ID card, International passport, O'level certificates etc. Fraudulent individuals have always tried to illegally replicate these documents using some computer technology such as Corel draw, Photoshop etc.

This action connotes a fraudulent act on the document fraud which happens to be a major challenge in almost all organizations today.

Document fraud can be seen as any alteration made to a document or any attempt to mimic an original document by any means. Document fraud can involve either false or altered documents, such as: certificates, passports and travel documents, visas, diplomas, apprenticeship or trade papers, certificates of birth, marriage, final divorce, annulment, separation, or death and police certificates.

Higher Education Degree Datacheck (HEDD) defines certificate fraud as crimes committed in relation to educational qualifications. It commonly falls under fraud, forgery, trade mark or copyright legislation (HEDD, 2016). Certificate fraud has eaten deep into all sectors in the Nigeria economy. Lagos State police arrested a banker with forged West Africa Examination Certificate (WAEC) result, Bachelor of Science (B.Sc.) degree certificate, NYSC discharge certificate (BellaNaija.com, 2015). A former Speaker of Nigeria's House of Representatives was impeached in 1999 after it was discovered that he falsified his age and forged his university certificate from Toronto University. A lecturer with forged academic certificates working with it for 12 years was arrested in the northern part of Nigeria. A confidential secretary at the National Broadcasting Commission who had worked for over 20 years was said to have forged a National Diploma (N.D.) certificate in Secretarial Studies. A senior medical officer at the federal ministry of health with a stolen medical license for over nine years was apprehended by the Nigeria police. The Department of State Security Services, DSS, Kwara State Command also reportedly arrested five persons, who allegedly specialize in forging academic certificates (Don, 2016). Even the current serving president of Nigeria was once accused of certificate fraud but his accusers could not prove he was guilty either because the manual system of verifying certificate failed or their accusation was wrong (PremiumTimes, 2016).

The manual system of verification is the most common type of verification system. In this system, the organization seeking to verify a document needs to send a delegate to the Institution for them to manually verify if the details contained in the document are accurate. Just to mention but a few because the cases of document fraud is endless, but we will narrow our scope down to Tertiary Institution Document fraud. In a survey covering the subject around one third of people admit to lying on their Curriculum Vitae (CV), with misinterpreting educational qualifications being the most common lie (HEDD, 2016). A recent analysis of 5,500 CVs by the Risk Advisory Group

(HEDD, 2016) found that 44% of CVs had discrepancies in education claims with 10% of those having false grades. This was an increase of 7% on the previous year.

There are several automated document verification systems in existence today, some schools have an online verification portal where organizations login to verify an applicant's document e.g. Lagos State University, West African Examination Council etc. Although this system is very accurate but how will organizations that have many applicants from different institutions verify the document of each of their applicants? They will need to open many websites in other to verify the documents; others may not even know the verification websites of the various schools. Some companies like ETX-NG CertVerify provides this service by standing as an intermediary between the organization seeking to verify the certificate of an applicant and giving a feedback within the space of 24hrs. Although this system solves the problem of visiting many websites but the time spent verifying document is a bit much.

A proposed system was built by a Master of Science (M.Sc.) student in Kenya, which requires all institution uploading the certificate details of each graduating student in their respective institutions to a centralized database, which will subsequently be used to verify the certificates of job applicants (Ochieng, 2016). Although this process is real time and accurate as the data stored in the centralized database are uploaded by the database admin of each institution but what happens if something happens to the database? The outcome will be disastrous as all the data may be lost or affected.

Educational establishments try to combat fraud and forgery in several ways, however, most of the methods on ground are time-consuming because they are manual, partly automated or involve human to human interaction. Hence this project attempts to design a cryptographic verification system to verify institutional documents and preserve the confidentiality of the information in them. The system is designed for employers or recruiters, organizations, graduates and institutions. The need to implement a verification system has been stressed in order to meet a digital growing nation with modern cryptographic technology and that is what this project research addresses.

1.2 Motivation

All documents or credentials that are printed are potentially subject to counterfeiting and forgery. Forgery can cause a lot of damage when it comes to trust and authenticity. The technologies that are put forth to stop or prevent forgery do not seem to be moving fast as the evolution of the forging techniques. There is a high market for forgery as well as opportunity with low cost, high quality results available. With respect to academic documents, further authentication problems include the variations from one school to the next, which causes consistency issues that can be taken advantage of, especially in international situations. A proper verification system would help foster and validate better the authenticity and integrity of institution documents.

1.3 Problem Definition

There are many instances of document fraud in Nigeria and the manual document verification system seems to be too slow and expensive as the cost of verifying the document is quite high and the process is tedious.

Over the years, most organization verify applicants' document manually by themselves. Some of the methods used by these organizations include checking the kind of paper used to print the document, checking the authenticity of stamp and seal on the document, comparing the applicants' document against previously verified documents from same institution. All these manual methods taken by these organizations have some major disadvantages as they led to document fraudsters improving their abilities to replicate these documents accurately and has also led to the employment of unqualified individuals occupying positions they do not merit. The process of verifying a degree certificate from Nigerian universities overtime is cumbersome and there exist no system that provides an integral interface for seamless verification.

There are several automated document verification systems in existence today, some were built to schools to verifying the authenticity of the documents awarded by that institution, many institutions across the world provides such service, an example of an institution with such facility in Nigeria is LASU.

As awesome as this seems, it's limited to only LASU, certificates from other schools cannot be verified on that web application, and only a very negligible amount of universities in Nigeria offers this service, even though all universities in Nigeria chooses to provide this service, it will still be stressful for the employer to verify the certificates of applicants from different universities due to the stress involved in remembering the Uniform Resource Locator (URL) address of each university and visiting different websites, take for instance there are over a hundred applicants from different universities in Nigeria, just imagine the stress and the time it will take to verify these certificates.

The prevalent rate of document fraud is globally on the increase as the population of fake academic documents by fraudsters cannot be tackled due to the problem of poor existing verification systems and the negligible interest exhibited by many employing organizations.

However, the problem of poor verification systems by academic institutions has given prevalence to the rise of academic certificate fraud and this is a challenge this research seek to address.

1.4 Research Aim and Objectives

The aim is to build a system for academic document verification, centered on integrating distributed databases for the purpose of providing a verification service where the following objectives are vital:

- To examine existing system with a view towards exposing their challenges.
- To propose a system capable of managing these challenges.
- Design and implement the proposed system.

1.5 Scope of Research

This research focuses on verification system for institution documents. It is centered on building a system that employs the cryptographic mechanisms to verify and validate the authenticity of documents claimed to have been issued by a certain institution.

1.6 Research Methodology

This research adopted the Microservices Architectural Design Pattern of Service Development which is a modern approach to building applications that acts as a collection of distributed, loosely coupled, small services that can communicate with each other, work independently.

For designs of the proposed system, well-defined activities using a combination of text and diagrams from the initial/conceptual idea to the actual physical design of the system application was modelled using Service oriented architecture Modelling Language (SoaML). Thereafter the designs were implemented into an actual working system. Implementation was done using the following set of tools: HTML, JavaScript and CSS as frontend, NGINX as middle-ware and Python and PostgreSQL database as backend and deployed finally on an AWS Cloud Server.

It thereafter concluded with summary, conclusion and recommendation.

1.7 Research Significance

This research provides a Document Verification Service for curbing document fraud where the concepts and knowledge exposed are beneficial to:

- Organizations seeking professionals and competent employees.
- Individuals seeking interest in verification systems.
- To reduce unprofessional individuals parading themselves with forged documents. E.g. in the medical profession, it can be very disastrous to employ an individual with a forged certificate.

1.8 Limitation of Study

This project has really motivated me in many ways that am so attached to it but my limitations are:

- Time: the data combined in the database of institution is not accurate due to time constraint, as time is needed to gather information from different departments and update the database accordingly.
- Finances.

CHAPTER TWO

LITERATURE REVIEW

In this section, the concept of document fraud was discussed, which includes the major reasons why individuals engage in fraudulent activities regarding academic, some of the adverse effects of academic documents forgery was also discussed as well as verification of academic documents. It went further to explore the various existing technique for verifying documents, the challenges faced in verifying the academic documents employing the respective techniques. Finally, it discusses the relevance of asymmetric cryptography – digital signature and Cloud computing technology in particular distributed computing.

2.1 Document Fraud: An Overview

Fraud is an intentional perversion of truth in order to induce another to part with something of value or to surrender a legal right. It can also be seen as a copy of something that is meant to trick people. Oxford dictionary defines Fraud as a wrongful or criminal deception intended to result in financial or personal gain.

The term 'fraud' commonly includes activities such as theft, corruption, conspiracy, embezzlement, money laundering, bribery and extortion (CIMA, 2008).

According to Article 204 of the Qatar Penal Code, forging a document means altering the truth thereof so that such alteration results in causing damage but with the intention of using it as a genuine document. Document fraud is no news to us any more as it has been a norm in our society today. A document is an official paper that gives information about something or that is used as a proof of something. Hence document fraud can be defined as a copy of an official paper that gives information about something with the intention of tricking people. There exist so many types of documents a few of these are degree certificates, birth certificates, O'level certificates, Identity cards, academic certificates, invoices, international passports etc. For the purpose of precision and effective study we will be focusing on Academic certificate fraud.

2.1.1 Academic Document Fraud

Every company needs to hire trained and academically qualified staff in order to provide efficient and quality service. There are quite a number of academic institutions in Nigeria today, from the primary level all through to the post-graduate level. All these institutions offer her pupil/student a certificate stating their performance while in the institution, these certificates include Primary School Certificate, West Africa Examination Council certificate, National Diploma certificate, Higher National Diploma certificate and Doctorate Degree certificate etc. A keen interest is on the tertiary institutions in Nigeria.

All tertiary institutions have a concise records of the honors awarded to her graduating student each year, these records are usually used to verify the authenticity of a claimed certificate by an individual or a job applicant when the need arises. This is due to the fraudulent activities by some unscrupulous individual parading themselves in forged certificates, working where they do not merit and causing havoc to the economy at large.

The offence of forgery includes any amendment by way of addition, deletion or otherwise in the writing of the instrument, figures, pictures or signs contained therein; putting any forged signature or false seal or altering any true signature, seal or signature of a person who is unaware of the true contents of the instrument; producing a counterfeit document or imitating an original; completing a document signed, sealed or stamped in blank without authority from the actual signatory or holder of the seal or the person making the thumb impression; falsely assuming the identity of another person or altering such identity in an instrument intended to be executed (Nizar, 2013).

Hence, Certificate Fraud can be seen as a criminal deception over a certificate with an intention to deceive. It can also be considered as an alteration to any document issued to an individual.

The 21st century has experienced a lot of cases involving fraudulent practices of academic certificates (Victor, 2014). Glasner (2015) supports this by taking note of the increase in the production of fake academic certificates all over the world. According to Grolleau et al., (2015), The United States alone was reported to have over two million fake certificates. Herly (2015) adds to this statement saying that between the range of 25 to 35 percent candidates in Australia have either falsified or exaggerated their academic credential to obtain employment.

There are many instances of certificate fraud in our country today, it runs through almost all sectors of the Nigeria economy, some instances have been earlier stated, other instances are as follows

The certificate forgery scandal rocking the senator representing Kogi West Senatorial District, Dino Melaye, who was accused of forging his degree certificates which he claimed to have gotten from Ahmedu Bello University (ABU) Zaria, his National Youth Service Corp Certificate and other academic certificates, the case was under investigation in 2017 but seems to have been swept under the rug (NewsPunch, 2017).

A female lecturer, by the name of Mrs. Asabe Garba, was dismissed after it was discovered that her master's degree from Ahmadu Bello University on the basis of which she was employed to teach at Ibrahim Badamosi Babangida University's Mass Communication department was forged (Azuka, 2014).

Minister of Aviation, Stella Odumah was accused of degree fraud, Twenty-four hours after her claim to have acquired a Master's degree from the St. Paul's College in the United States was punctured, PREMIUM TIMES has uncovered the false claim by our embattled Minister of Aviation, this time, the minister lied on oath to the Nigerian Senate, claiming another American 'university', Pacific Christian University awarded her an honorary doctorate degree in 1998. But investigations by PREMIUM TIMES suggest that no university called Pacific Christian University exists in the United States (Mojeed, 2014).

Sahara Reporters has obtained a document issued by WAEC declaring that Senator Emmanuel Nnamdi Uba, forged both a secondary school certificate and "Confirmation of Result" which he presented to British authorities. He also claimed that he obtained bachelors, masters and doctorate degrees from Concordia University, Canada, California State University, and Buxton University in the UK. Investigations disclosed that Mr. Uba indeed registered as an undergraduate student in both universities, but that he dropped out of both institutions without completing enough work to earn even a bachelor's degree, further investigation also showed that Buxton University is not an accredited institution of learning, but a "certificate mill" that sells degrees to anybody willing to pay a small fee. The fake institution does not have any physical address in Britain. (Sarahareporters, 2017)

There is no genuine reason behind fraud and any explanation of it needs to take amount of various factors. Looking from the fraudster's perspective, the major drive to document fraud is the desire to obtain employment, obtain promotion coupled with the poor system of document verification and even the lackadaisical attitudes exhibited by some organizations to verify documents. Glasner (2015) supports this by noting that job applicants forge academic documents in other to get an edge in the competitive market, even Garwre (2015) agrees that the forgery is motivated by unemployment rate in most countries which makes job employers consider candidates with higher academic qualifications.

2.1.2 Adverse Effect of Fraud

While only relatively few major frauds are picked up by the media, huge sums are lost by all kinds of businesses and organizations as a result of the high number of smaller frauds that are committed. Surveys are regularly carried out in an attempt to estimate the true scale and cost of fraud to business and society. According to (CIMA, 2008) Findings varies, and it is difficult to obtain a complete picture as to the full extent of the issue, but these surveys all indicate that fraud is prevalent within organizations and remains a serious and costly problem. The risks of fraud may only be increasing, as we see growing globalization, more competitive markets, rapid developments in technology, and periods of economic difficult.

Among other findings, the various surveys highlight that organizations may be losing as much as 7% of their annual turnover as a result of fraud, corruption is estimated to cost the global economy about $1.5 trillion each year, only a small percentage of losses from fraud are recovered by organizations, a high percentage of frauds are committed by senior management and executives, greed is one of the main motivators for committing fraud, fraudsters often work in the finance function, fraud losses are not restricted to a particular sector or country, the prevalence of fraud is increasing in emerging markets. (CIMA, 2008). In support of this, Glasner (2015) noted that companies in the United States lose up to $50,000 and loss of productivity with a single employee with a forged academic certificate. Additionally, forgery of academic certificates devaluates the institution where the qualification is pupated to be obtained and also ruins the reputation of the legitimate students of the said institution (Garwe, 2015).

Academic document forgery therefore poses a big challenge to the economy of a country and also the integrity of both the academic institution and their legitimate document holders as earlier stated. Hence a need for an academic document verification system arises.

2.2 Documents Verification

According to the research conducted by Eldefawrey M. H (2012), all documents or credentials that are printed are potentially subject to counterfeiting and forgery. Forgery can cause a lot of damage when it comes to trust and authenticity. Verification is the process of determining or confirming that someone (or something) is original. Documents Verification one the other hands can be defined in various ways as the process of proving the correctness or authenticity of a document by using a proven method or technique.

Ravinshankar S. (2009) defines documents verification as the process of ensuring that documents received from holder are genuine and that the holder is the rightful owner. The problem is that the verification of certificates is costly and time consuming using the traditional methods in which the person to verify calls the issuing institute to make sure that the said document is correct and that the information is real.

The main aim of document verification is the ability to trace the origins of a document to a specific person, the device that produced it or the place where it was produced (Mantoro, 2015). Forgeries poses a huge threat to the integrity of documents, with significant dangers in terms of authentication and trust. It is therefore important to protect the integrity of a document to prevent problems arising from the modification of a document by intruders (Mantoro, 2015). Technologies that are put forth to stop or prevent forgery do not seem to be moving as fast as the evolution of the forging techniques (Husain, 2014).

With respect to academic documents, further authentication problems include the variations from one school to the next, which causes consistency issues that can be taken advantage of, especially in international situations (Eason, 2018). There are two basic institution document categories that are considered in this project; digital based documents and paper-based documents or printed documents. Almost all the documents can be handled in a digital manner except for a few, for now, certificates. The reason for the exception is that all digital documents are easy to forge without leaving any clues. Furthermore, the prevalence of forged certificates results from the increased global demand for higher education, which exceeds the university capacity of the world.

According to the research conducted by Chen Z (2018), there are two main types of forgery, type 1 and type 2. Type 1 forgery is when some part of the original document is changed in order to benefit someone who was not benefitted by the original document. In this case, the base substance, normally the paper remains legal and valid, but the information that is contained therein if forged. The second, type 2 forgery is when both the base substance and the information contained therein in fake.

There are three entities that must be present to accomplish the process of document verification which are the issuer, the owner and the verifier. The issuer represents the entity that issues the document, in our case, an Institution. The owner represents the person who owns the document. The verifier represents the employer or third party that verifies the document (Chen-Wilson, 2010).

Institution based documents categorized as paper based and digital based documents still commonly used today are briefly explained as follows:

Paper Based Document: There are many types of paper based documents such as graduation certificates, birth certificates, etc. The information inside the paper based documents are subject to threats like forgery; despite measures taken to protect them from attacks. Husain A. (2014) attributes that to the lack of verification. These paper based documents despite being widely used can be damaging. The most important disadvantages are 1. Slow to retrieve. 2. It can eventually consume physical space. 3. It can be costly if changes are required on the document; for example a faulty name was printed, more papers would have to be used and that extra cost for the entity issuing the document; this indirectly also affects the environment. 4. They can be easily lost.

Despite these drawbacks with Paper based documents, it also avails itself with advantages and as well it is still very much commonly used.

Digital Based Document: These being documents issued in a digital form. It is usually issued through a secure certification and verification method (Chen-Wilson, 2010). It is mostly adopted in order to solve the management problems of paper based documents. However one of the important reasons why digital documents are widely adopted is that digital documents provide a unique feature which is portability; it is easy to transfer documents when they are digital. They are considered environmental friendly and can easily be organized without taking much space. Although, in their simplest form is the easiest to forge without the need for special hardware.

Despite the advantages of digital based documents, they are not widely spread as the paper based documents and are not the preferred method for many institutions. Even if digital based documents are issued, paper based documents are still required and needed.

2.3 Document Verification Processes and Technologies

There are several incidents where fake documents were discovered. As a result, many techniques arose like holograms, stamps and wet-signatures (Ravinshankar, 2009). However, these techniques can easily be replicated to create forged documents. Fortunately, there are continuous researches aiming to provide newer and better means to authenticate, validate, and verify the paper based documents and also digital based documents.

Signature extraction is a technique applied to verify bank cheques (Madasu, 2018). Signatures are of great importance in any paper based document. The availability of signature reflects the authenticity of document and level of authority that handled the document. In the research conducted by (Madasu et al, 2018; Zhu et al, 2008), they focused on the signature verification only and they targeted bank cheques. Their proposed system utilizes their signature database for verification. Given this project is focusing on Institution documents, this solution is not the best solution to adopt for institution documents simply because 1) using the signature on its own would not ensure that the certificate is not forged and that the information is correct; the signature merely reflects that the document has been authorized by a specific person. This does not ensure in any way that the content is valid and is correct. Another issue with this approach is that 2) it still can be forged. Anyone can just copy the original signature and with the right tools a new document is created. Documents can be forged and the same signature used; the system will simply identify the signature and ensure that it exists in the database of signatures. The third issue with the proposed approached is that the technique was proposed with bank cheques in mind. Bank cheques have specific formats and templates than institution documents.

Print signature is another technique which was proposed to verify documents (Zhu et al, 2008). Print signature has two procedures which are the 1) registration and 2) authentication. The registration procedure is given to the document to be protected; and the authentication procedure is to verify the authenticity and originality of the printed document. The verification process is made based on Optical Character Recognition (OCR) techniques. The OCR technique is made by fetching the features of document to be verified and then matched with saved features in database or somewhere else. However, it does not verify the source or where it was issued from. Saving features on a database may not be the safest approach. If the database is exposed the verification process will fail. The best approach is that if an algorithm verifies on the fly without the need to store the sensitive details needed to verify on the database.

The use of hash value for verification was proposed by the researchers (Voloshynovskiy et al, 2015). The proposed method in the conducted research contains two stages which are document enrollment and authentication stage. They proposed to compute a hash value in the document enrollment stage and then store it in a hash value database. The document is authenticated through computing the hash value in the database. If it matched, the document is authentic and if it doesn't, the document is not authentic. The generation of hash value depends on the content in the certificate which will help in protecting the document from any changes. However, this approach does not consider securing the stored hashed value from leakage or being attacked.

Barcode is one of the most well-known approaches which can be used for document verification. The barcode can be analyzed to obtain the hidden data. In term of encoding type, the barcode can be classified into two categories, namely, are 1D barcode and 2D barcode. The 1D barcode usually consists of varying of parallel lines different in both widths and space. Barcode offers several benefits such as High data capacity; error correction ability; and no additional storages (Lin et al, 2013). 2D barcode can be used to hold information in both sides horizontally and vertically. Its content can be recovered reliably by scanning and decoding of the barcode. The main use of 2D barcode is to hold significant amount of data, typically of the order of 500 bytes per square inch. One of the main advantages of 2D barcode is that it can be printed on paper by normal printed and scanned by normal scanners (Adjei, 2013). For document verification, they proposed an application which scans the 2D barcode and the document image to process it. The application reads the document image line by line utilizing OCR technique to recognize the text in the image of the document taken. Their proposed process to decrypt the information from the 2D barcode.

The researchers (Salleh et al, 2015) suggested somewhat similar verification approach to the researchers (Lin et al, 2013). They proposed to embed the hashed unique key generated from the timestamp, track number and the content of document into 2D barcode. For document verification, they proposed to scan the document using OCR techniques. The text of the document along with the timestamp and the tracking number will be extracted from the 2d barcode. However the extracted data is hashed using hashing algorithm. The generated hash value is matching with the hash value stored in 2D barcode. If the hash values are same, the document is original else the document is fake. The proposed method is adopted within same organization.

Similar to the verification approach which was proposed by (Salleh et al, 2015), the researcher (Eldefrawy, 2016) also proposed to use 2D barcode. However their intention was to verify documents sent from point A to point B with a known sender and receiver. The verification method is proposed based on a scenario where user A wants to send document to user B and user B wants to verify the sent document sent by A. Hence, he proposed that the important parts of the document such as time stamp, issuing number, and sender signature and hashing value of the content of the document be embedded in the 2D barcode. For document verification, the receiver decrypts the content through scanning the 2D barcode. This approach added extra details to the embedded data in the bar code however these details may not be needed in institution documents and it suffers similar drawbacks to the previous approach mentioned.

RFID is another technique used for document verification. RFID stands for Radio Frequency Identification; it uses the radio frequency waves to transfer data. The data is transferred between two entities which are the reader and moveable item. The moveable item is tagged to categorize, identify and track it (Voloshynovskiy et al, 2005). In the research conducted by (Mudraganam, 2009), they developed smart degree system based on RFID. Their system utilized the fingerprint of graduates for the purpose of document verification. They proposed that the university issues the documents with RFID tags; this tag contains the important information related to the graduate like name, graduation date, the program, the degree and biometrics (fingerprint) of the graduate. For certificate verification, an interrogator must be used to read the embedded data in the tag and also the verifier have to log into a website which is usually indicated in the back of the document, and download the required software to enable interpreting the tag. The issue with the proposed solution is that the process of authentication is time consuming, especially that the external entities that might verify the document will have to download a software in order to read the encrypted data along with the need to have an RFID reader to read the tag and download its data. Another major problem associated with this approach is that the documents mailed or copied cannot be verified since they will lack the RFID in them. The original document has to be present and this is not a feasible solution making RFIDs useless for institution documents.

Another dominant approach utilized in document verification is digital watermarking (Garain et al, 2008; Sheng, 2012). It is a technique that can be utilized for protecting the copyrights or ownership; it also can be used to prevent forgery in printed documents. It is a method that can be used to embed some information in the cover image. This information can later be extracted and processed. The way watermarking works in the simplest form is through embedding the main information of document as a watermark in the same document. For document verification, the hidden information can be extracted using certain algorithms (Afrakhteh, 2015). Usually it is very hard to notice the embedded watermark by the human naked eyes, hence if tools like OCR is used to read the contents of the watermark would be lost. The drawbacks of using watermarking is that it can easily be lost when transforming paper-based to digital and vice-versa. Also to add it requires change in the type of paper used for documents as well as the process of generating the document. As for digital watermarking, institution documents are not all digital and still are being paper-based and so is most of all real-life documents.

2.4 Verification Approach and Technology

Certain technologies in the past have been tried and used in the context of documents verification to combat fraud in the society. However, with the trend of more technologies in recent time including mathematical algorithms like cryptography and the new architecture called microservices proposed for building complex applications, as well as the cloud computing platforms that have made continuous integration and continuous delivery faster and iterative, this several technologies are discussed here.

2.4.1 Asymmetric Cryptography

Asymmetric cryptography, or Public-key cryptography, is a cryptographic system that uses pairs of keys: public keys (which may be known to others), and private keys (which may never be known by any except the owner). The generation of such key pairs depends on cryptographic algorithms which are based on mathematical problems termed one-way functions. Effective security requires keeping the private key private; the public key can be openly distributed without compromising security (Tunggal, 2020).

In such a system, any person can encrypt a message using the intended receiver’s public key, but that encrypted message can only be decrypted with the receiver’s private key. This allows, for instance, a server program to generate a cryptographic key intended for a suitable symmetric-key cryptography, then to use a client’s openly shared public key to encrypt that newly generated symmetric key. The server can then send this encrypted symmetric key over an insecure channel to the client; only the client can decrypt it using the client’s private key (which pairs with the public key used by the server to encrypt the message). With the client and server both having the same symmetric key, they can safely use symmetric key encryption (likely much faster) to communicate over otherwise-insecure channels. This scheme has the advantage of not having to manually pre-share symmetric keys (a fundamentally difficult problem) while gaining the higher data throughput advantage of symmetric key cryptography (Bernstein, 2020).

With public-key cryptography, robust authentication is also possible. A sender can combine a message with a private key to create a short digital signature on the message or document file. Anyone with the sender’s corresponding public key can combine that message with a claimed digital signature; if the signature matches the message, the origin of the message is verified (i.e., it must have been made by the owner of the corresponding private key) (Menezez, 2016).

Public key algorithms are fundamental security primitives in modern cryptosystems, including applications and protocols which offer assurance of the confidentiality and authenticity of electronic communications and data storage. Some public key algorithms provide key distribution and secrecy, some provide digital signatures (e.g. Digital Signature Algorithm), and some provide both (e.g. RSA). Compared to symmetric encryption, asymmetric encryption is rather slower than good symmetric encryption, too slow for many purposes. Today’s cryptosystems (such as TLS, Secure Shell) use both symmetric encryption and asymmetric encryption.

Before the mid-1970s, all cipher systems used symmetric key algorithms, in which the same cryptographic key is used with the underlying algorithm by both the sender and the recipient, who must both keep it secret. Of necessity, the key in every such system had to be exchanged between the communicating parties in some secure way prior to any use of the system – for instance, via a secure channel. This requirement is never trivial and very rapidly becomes unmanageable as the number of participants increases, or when secure channels aren't available, or when, (as is sensible cryptographic practice), keys are frequently changed. In particular, if messages are meant to be secure from other users, a separate key is required for each possible pair of users (Mavroeidis, 2018).

By contrast, in a public key system, the public keys can be disseminated widely and openly, and only the corresponding private keys need be kept secret by its owner.

Two of the best-known uses of public key cryptography are:

- Public key encryption, in which a message is encrypted with a recipient's public key. For properly chosen and used algorithms, messages cannot in practice be decrypted by anyone who does not possess the matching private key, who is thus presumed to be the owner of that key and so the person associated with the public key. This can be used to ensure confidentiality of a message.
- Digital Signatures, in which a message is signed with the sender's private key and can be verified by anyone who has access to the sender's public key. This verification proves that the sender had access to the private key, and therefore is very likely to be the person associated with the public key. This also ensures that the message has not been tampered with, as a signature is mathematically bound to the message it originally was made from, and verification will fail for practically any other message, no matter how similar to the original message.

One important issues is confidence/proof that a particular public key is authentic, i.e. that it is correct and belongs to the person or entity claimed, and has not been tampered with or replaced by some (perhaps malicious) third party. There are several possible approaches, including:

A public key infrastructure (PKI), in which one or more third parties – known as certificate authorities – certify ownership of key pairs. TLS relies upon this. This implies that the PKI system (software hardware, and management) is trust-able by all involved.

A "web of trust" which decentralizes authentication by using individual endorsements of links between a user and the public key belonging to that user.

2.4.2 RESTful Web Services

Representational State Transfer (REST) is a client-server architectural style that provides a behavioral model for client applications and web services. REST describes a number of design principles and constraints, such as stateless communication and the use of uniform interfaces and self-descriptive messages that should be applied in REST-based services (Fielding, 2015).

Services developed according to the REST style are usually referred to as RESTful web services. Each RESTful web service should expose a set of resources, i.e., any type of information exclusively identified by a Uniform Resource Identifier (URI) that can be referenced through the web. The access to a resource occurs through a uniform interface that provides standard access methods for handling the resource. When a particular resource is accessed by a client application, a representation of this resource reflecting its current state should be returned in response to the request. The decoupling of a resource from its representation enables a service provider to return different representations for the same resource in order to provide enhanced flexibility for different client applications. In general, resources are represented in either Extensible Markup Language (XML) or JavaScript Object Notation (JSON) formats.

RESTful web services provide uniform interfaces accessed only through HTTP methods, so that interoperability among different web services can be accomplished, and the development of client applications to interact with RESTful web services is simplified. Moreover, REST-based technologies provide a simple, lightweight interaction model and have been increasingly used in the development of web services.

2.4.3 Microservices

Microservices – a variant of the service-oriented architecture structural style – arranges an application as a collection of loosely-coupled, independent, distributed services. In a microservice architecture, services are fine-grained and the protocols are lightweight.

Microservices are increasingly used in the development world as developers work to create larger, more complex applications that are better developed and managed as a combination of smaller services that work cohesively more extensive, application-wide functionality.

Speed, agility, lower costs and little impact in the legacy infrastructure are some of the most important characteristics of a microservices architecture, which further distinguishes it from Monolithic architecture. It allows services to evolve independently, works with different technologies and frameworks and makes testing and debugging much easier.

2.4.4 Cloud Computing

Cloud Computing according to IEEE Journals & Magazine, is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user (Montazerolghaem, 2020). The term is generally used to describe data centers available to many users over the Internet.

Steve Ranger (2018) defines cloud computing as the delivery of on-demand computing services – from applications to storage and processing power – typically over the internet and on a pay-as-you-go basis. Rather than owing their own computing infrastructure or data centers, companies can rent access to anything from applications to storage from a cloud service provider.

One benefit of using cloud computing services is that firms can avoid the upfront cost and complexity of owning and maintaining their own IT infrastructure, and instead simply pay for what they use, when they use it. In turn, providers of cloud computing services can benefit from significant economies of scale by delivering the same services to a wide range of customers.

Cloud computing services cover a vast range of options now, from the basics of storage, networking, and processing power through to natural language processing and artificial intelligence as well as standard office applications. Pretty much any service that doesn't require you to be physically close to the computer hardware that you use can now be delivered via the cloud. It underpins a vast number of services. That includes consumer services like Gmail or the cloud back-up of the photos on your smartphone, through to the services which allow large enterprises to host their data and run all of their applications in the cloud. Netflix relies on cloud computing services to run its video streaming service and its other business systems too, and have a number of other organizations.

Cloud computing is becoming the default option for many apps: software vendors are increasingly offering their applications as services over the internet rather than standalone products as they try to switch to a subscription model (Simpson, 2017). However, there is a potential downside to cloud computing, in that it can also introduce new costs and new risks for companies using it.

A fundamental concept behind cloud computing is that the location of the service, and many of the details such as the hardware or operating system on which it is running are largely irrelevant to the user (Oestreich, 2015). It's with this in mind that the metaphor of the cloud was borrowed from old telecoms network schematic, in which the public telephone network (and later the Internet) was often represented as a cloud to denote that it just didn’t matter – it was just a cloud of stuff. This is an over-simplification of course; for many customers location of their services and data remains a key issue.

2.5 Summary

Based on what has been presented in the previous section, there are many techniques proposed for paper based document verification. Most of these techniques require change in the process of document generation either by changing template, changing paper, changing printers, adding hardware or even adding extra information. This change may mean that the institution or verifier need the proper knowledge to execute and run the proposed technique. This also means that older documents may not work with the new introduced techniques. To also add, some proposed techniques require a change that is not always easy or cheap like in creating a third body to verify documents.

As reflected some techniques are mostly suitable for specific domain and document. Others were proposed based on specific environments and conditions like environments that assumes both sender and receiver are known to each other.

This chapter highlighted the challenges of several approaches to institution documents verification system as a major factor affecting institutions or foreign bodies keen on the verification of documents presented to them by prospective candidates. Thus, a motivation for the aim of this study to develop a Cloud based Verification System using Digital Signatures and QR code based on Microservices design pattern to facilitate verification process from anywhere in the world. The analysis and designs are postulated in the next chapter.

CHAPTER THREE

SERVICE SYSTEMS ANALYSIS AND DESIGN

This chapter gives an overview of the methodology in engineering the system. To build the system given the microservice nature, the Service-oriented Heterogeneous Architecture and Platform Engineering (SHAPE) approach was adopted. This approach was significant as it is most suitable for building cloud-based applications (Elvesaeter, 2011).

The approach first considers a system as a collection of independent loosely coupled distributed services knitted to achieve a common purpose. It first sees a system as a collection of services that need analysis, then proposes an architecture and models design of the system architecture using Service Oriented Architecture Modelling Language (SoaML). Thus, the remaining sections of this chapter focuses on its Analysis phase to Designs.

3.1 Service System Analysis

Services in this context are independent, loosely coupled, easy to maintain and test functionalities that are structured as an architectural style to form a whole application.

In the analysis phase, the requirements of the microservice are further refined and translated into conceptual models by which a technical development team can understand. It is also in this phase that an architecture analysis is done to define the high-level structure and identify the Microservice interface contracts.

3.2 Analysis of Existing System

In this research, the University of Benin (UNIBEN) was selected as the system under study. The institution/organization seeking the verification of a document will have to travel down to the university or alternatively send a written request to the school to verify the result. This request will be addressed to the office of the Registrar, who in turn sends the letter to the Senate matters division in Exams and Record Department. The senate matters division has a copy of the Brochure awarded to students yearly on graduation, the officer in charge will then check the Brochure and check if the applicants name is in the Brochure, using the following constraints (Year of graduation, Class of Honor, Full name). This will enable the officer in charge to find the students information in the barest possible time, but in a case where the student’s information is not found in the Brochure, the officer will go further to check the senate approved result for that session, the session approved result contains the necessary details in the document of the graduating student for each session, if the students' name is still not found, the results sent to senate for approval will be checked if the student is among those whose names were struck out due to the fact that they are still having one or two outstanding courses as referred. Hence, the unit will reply the organization requesting for document verification stating if the certificate is authentic or not with reasons.

3.3 Constraints of the Existing System

The existing system which is a manual system of student verification is associated with the following constraints or problems;

- Unreliability: this system (the existing system) can be manipulated since human beings are directly involved in the process of verification.
- Human Error: such as an oversight can lead to wrong conclusion as to if a certificate is authentic or otherwise because humans can get tired or fatigued. Human can be bribed, deceived and can show favoritism.
- Speed: this has to do with time; it takes longer time to send an application letter for verification and the reply thereof.
- It increases paper work since it is not an electronic system.
- During the process of application for verification and reply to application, some important document might be lost.
- Un-confidentiality: privacy is not considered in the manual system, important and private information may be made open during the process of verification, as many people (more than two persons) are involved.

Having investigated and identified several weaknesses, the above defects limits the maximal utilization of the manual system, thereby a need for a computerized system arises.

3.4 Overview of the Proposed System

The proposed system is called DocVerify. DocVerify is a cloud based software application which uses a web service technology to provide verification service on documents of graduates for schools in Nigeria, each school having her own separate repository where the admin is software admin is supposed to upload the document details of every new graduates.

The processing logic for this system is written in Python and resides on an Amazon Web Services (AWS) EC2 Instance (or Virtual Machine), which also interacts with a relational database (PostgreSQL) to store records.

This system uses a mechanism known as **Asymmetric Cryptography or Public Key Infrastructure (PKI)**. This mechanism is used in several established platforms such as Github, AWS, Digital Ocean, and it is as well behind the handshake protocol of HTTPS. It is cryptographic system that uses a pair of keys known as **public keys** and **private keys.** A private key can be used to sign a message or document file. A sender can combine a message with a private key to create a short digital signature on the message or document file. Anyone with the sender's corresponding public key can combine that message with the claimed digital signature; if the signature matches the message, the origin of the message is verified.

DocVerify works in the sense that the heart of the system will be the institution's private key. For a particular institution that chooses to adopt this system, they become the admin of the system for that institution. DocVerify generates a private key for the admin of the particular institution on board. The heart of the system is the school's private key. Because the admin will use that private key to digitally sign documents. The institution is in charge of how to administer their public key infrastructure to distribute the admin's public keys so other authorities can verify it. There are several PKIs companies to help store and distribute public key. Thus, the parties that want to be involved with the verification of documents would have already obtained a public key from the admin or our PKI.

The steps needed to carry out the operation of document verification are as follows:

- ➢ The clients which are bodies e.g. WES, that are interested in verification of a particular document of a candidate packages the document into a POST request message containing the document, a signature and a public key.
- ➢ The REST request is sent to the web service as the body of an HTTP POST request.
- ➢ This triggers the backend of DocVerify which uses asymmetric cryptography to first ascertain that the public key sent along with the request is associated with the admin's private key.

- It also checks to verify that if indeed the corresponding private key (the admin) associated with the received public key was the key used to digitally sign the document received and ensures the signature produced is the same with the one received.
- The application processes all this as required and responds with a response of either STUDENT DATA or INVALID if signature is recovered.

It is quite obvious that this system provides an integral interface for seamless verification of documents. It should be noted that the response to the HTTP request can be consumed using any language of choice, hence this service is language independent since a service writing with Python can be consumed using Java, PHP, Nodejs or Golang etc. Since this proposed system is web service powered, it can run any operating system hence it is operating system independent.

Since this system is web-based, the cost of carrying out manual verification which is the normal practice in the University of Benin will be reduced.

DocVerify was built to have a distributed repository in order to avoid failure in the overall system in a situation when one of the database is being maintained and to avoid loss of all data if one of the database gets damaged unlike the centralized database.

DocVerify offers that all documents printed out by admins are followed with QR codes embedded on the page which contains a verification link that was generated when the document was digitally signed by the admin. This allows for paper based documents to be verified too by scanning the QR code from the paper document and quickly verifying the document straight on the browser.

The human limitation of oversight will be reduced to an insignificant level because it’s a computerized system and a computer cannot get fatigued, the system runs 24/7 which is not possible in the manual system.

3.5 Service-oriented Engineering Process Model

The engineering approach used in this work is that of **Software-oriented Heterogeneous Architecture and Platforms Engineering (SHAPE)** process model prototyping, this model was chosen because it bests models SoaML and makes SoaML dominant for building components as services.

Heterogeneous Architecture requires programs to be written differently for each of the dissimilar instruction sets. The goal is to offer substantially better performance or cost by devoting the appropriate parts of the application to machine designs optimized for specific types of computing.

Platform Engineering is about writing code that bridges the gap between software and hardware and testing the system so that it runs effectively and smoothly. It helps to reduce engineering friction which, in turn, allows fast value delivery to customers. Efficient platform architectures exist at the intersection of Engineering, Industry and People, balancing each one's goals in service of facilitating rapid value delivery to clients.

Mainly SHAPE principles hold well with the DevOps environment, which involves a delivery cycle that includes preparing, designing, testing, deploying, releasing and monitoring with active coordination between various team members. In short, SHAPE's core concepts are the flexibility of implementing changes, automation, continuous delivery, and fast feedback reaction. SHAPE's blueprint facilitates continuous innovation, automation, increased productivity and reduced time to market.

Good quality and pace are important consideration for building platforms and hence the need for SHAPE.

Hence, Software-oriented Heterogeneous Architecture and Platform Engineering process model saves me time in the development of this services.

3.6 Service Systems Architecture

Service architecture provides an explanation of how systems behave on a structural level. The systems have a collection of components that are designed to accomplish a specific task or set of tasks.

Architecture in general is a good way to figure out a software, although there are many architectural design patterns but a selected few has been selected to design the service of this application because it best fits the modelling of microservice-based applications. It is adequate for microservices applications and familiarity with the design patterns are considered. One designs pattern covered here is the Hexagonal Architecture Diagram.

3.6.1 Hexagonal Architecture Diagram Design

Hexagonal Architecture, or to call it properly, Ports and Adapters, is driven by the idea that the application is central to the service. It is an architectural pattern used in software design aiming at creating loosely coupled application components. All inputs and outputs reach or leave the core of the application through a port that isolates the application from external technologies, tools and delivery mechanics.

Hexagonal Architecture draws a thick line between the software's inside and outside parts, decoupling the business logic from the persistence and service layer. The inside part makes up the use cases and the domain model it's built upon. The outside part includes UI, database etc. The connection between them is realized via ports and their implementation counterparts are called adapters. In this way, Hexagonal Architecture ensures encapsulation of logic in different layers, which ensures higher testability and control over the code.

3.6.1.1 Why Hexagonal Architecture?

The main idea of Hexagonal Architecture is decoupling the application logic from the inputs and outputs and this makes it perfect for microservice based applications. It helps to free the most important code of unnecessary technical details and to achieve flexibility, testability and other important advantages that make working process more efficient.

As it facilitates the detachment of external dependencies, it helps with the classes that we anticipate will be swapped out in production in future and the classes that are intended to be faked in tests.

- High Maintainability, since changes in one area of an application doesn't affect others.
- Ports and Adapters are replaceable with different implementations that conform to the same interface.
- The application is agnostic to the outside world, so it can be driven by any number of different controls.
- The application is independent from external services, so you can develop the inner core before building external services, such as databases.
- Easier to test in isolation, since the code is decoupled from the implementation details of the outside world.

Each side of the hexagon represents an input – port that uses an adapter for the specific type. Hence, ports and adapters from two major components of Hexagon Architecture:

3.6.1.2 Ports

A port is a gateway, provided by the core logic. It allows the entry or exiting of data to and from the application. The simplest implementation of a Port is an API layer. Ports exist in 2 types: inbound and outbound.

An inbound port is the part of the core exposed to the world that defines how the Core Business Logic can be used.

An outbound port is an interface the core needs to communicate with the outside world.

3.6.1.3 Adapters

Adapter transforms one interface into another, creating a bridge between the application and the service that it needs. In hexagonal architecture all communication between primary (which use system to achieve a particular goal) and secondary actors (which system uses to achieve primary actor's goals) and application ports is done with the help of adapters.

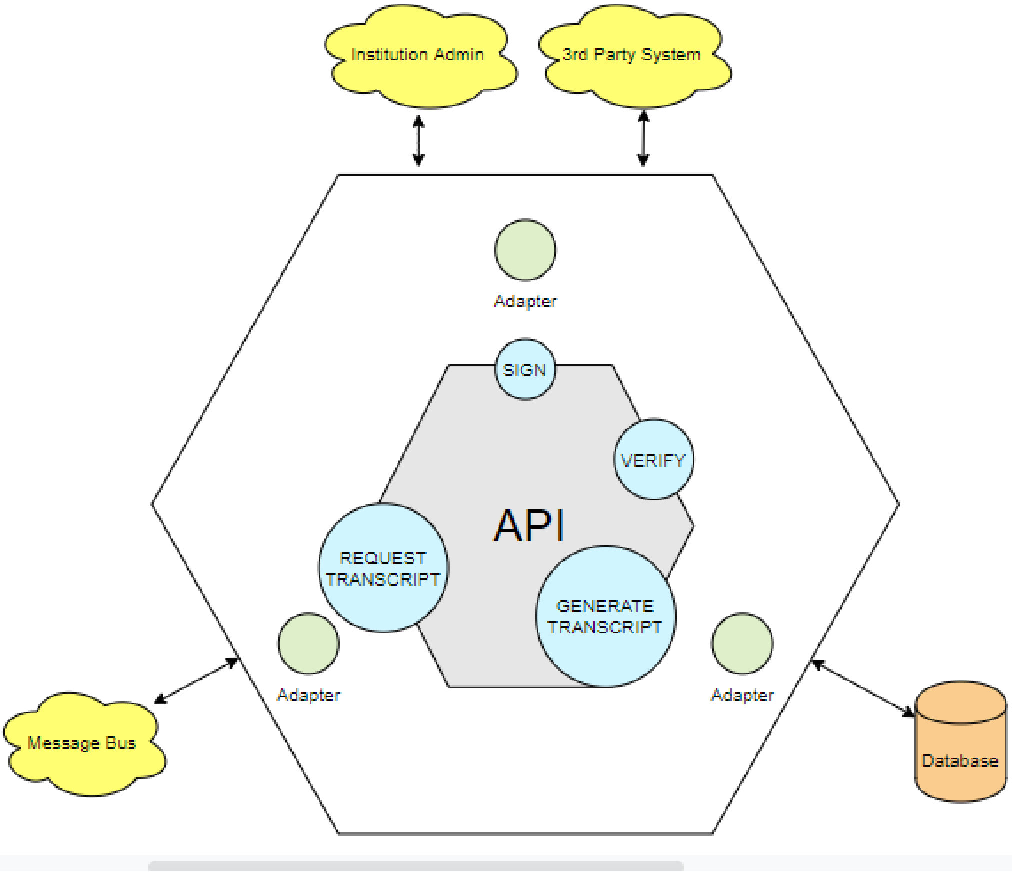


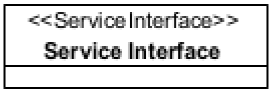
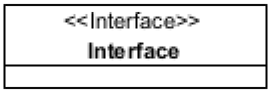
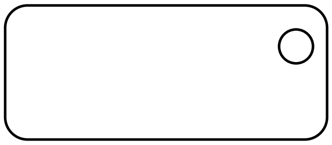
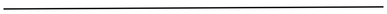
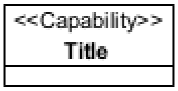
Fig 3.1 System Architecture for DocVerify

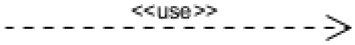
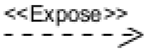
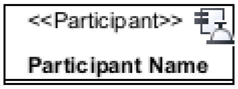
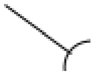
3.7 Service System Design - SoaML

The Service oriented architecture Modelling Language (SoaML) specifies services including the functional capabilities they provide, what capabilities consumers are expected to provide, the protocols or rules for using them, and the service information exchanged between consumers and providers. SoaML is an Object Management Group (OMG) standard that profiles a domain neutral modelling language to architect and model microservices with the use of Unified Modelling Langauge (UML). It identifies services, the requirements they are intended to fulfil, and the anticipated dependencies between them. It provides the ability to define classification schemes having aspects to support a broad range of architectural, organizational, and physical partitioning schemes and constraints.

SoaML profile is organized into five SoaML diagram types, namely, the Service Interface Diagram, Service Participant Diagram, Service Contract Diagram, Service Architecture Diagram and Service Categorization Diagram. Each of them provides a unique view to describe and help to understand services and services architecture.

Table 3.1 SoaML Notations

Name	Representation	Description
Service Interface		A ServiceInterface defines the interface and responsibilities of a participant to provide or consume a service. A ServiceInterface is the means for specifying how a participant is to interact to provide or consume a Service.
Interface		Simple interfaces define one-way services that do not require a protocol. Such services may be defined with only a single UML interface and then provided on a "Service" port and consumed on a "Request" port.
Role		It defines specific roles for each participant plays in the service interaction. These roles have a name and an interface type.
Connector		Connect roles in a service interface.
Capability		A Capability models the ability to act and produce an outcome that achieves a result that may provide a service specified by a ServiceContract or ServiceInterface irrespective of the

		Participant that might provide that service.
Usage		A ServiceInterface specifies its required needs through Usage dependences to Interfaces.
Expose		The Expose dependency provides the ability to indicate what Capabilities that are required by or are provided by a participant should be exposed through a Service Interface.
Participant		A Participant represents some (possibly concrete) party or component that provides and/or consumes services (participants may represent people, organizations, or systems that provide and/or use services).
Port		A port is the feature that represents the point of interaction on a Participant where a service is actually provided or consumed.

3.7.1 Service Interface Diagram

Service interface allows for bi-directional services. Such service involves the communication between provider and consumer of services in completing services. Service contract defines how participants work together to exchange value and we will talk about it when introducing the service contract diagram.

Service interface diagram allows for the modeling of service specification. You can model simple interfaces and service interfaces in service interface diagram.

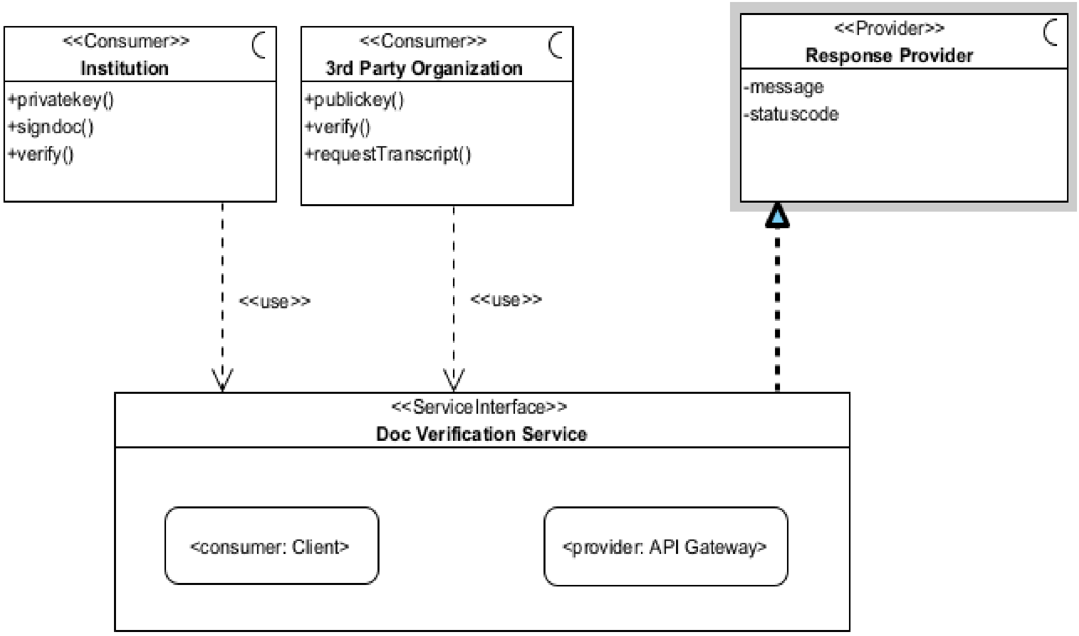


Fig 3.2 Service Interface Diagram for DocVerify

3.7.2 Service Categorization Diagram

In order to allow model elements to be used for multiple purposes and to be viewed from different perspectives, we need a way to organize the content of model. Categorization is available for such purpose.

The SoaML service categorization diagram allows categorizing SoaML elements with catalog, categories and category values.

3.7.3 Service Participant Diagram

In SoaML, participant represents certain party or component that provides and/or consume services. Participants can be software components, organizations, system or individuals.

Service participant diagram allows for modeling primarily the participants that play roles in services architectures. It also presents the services provided and used by these participants.

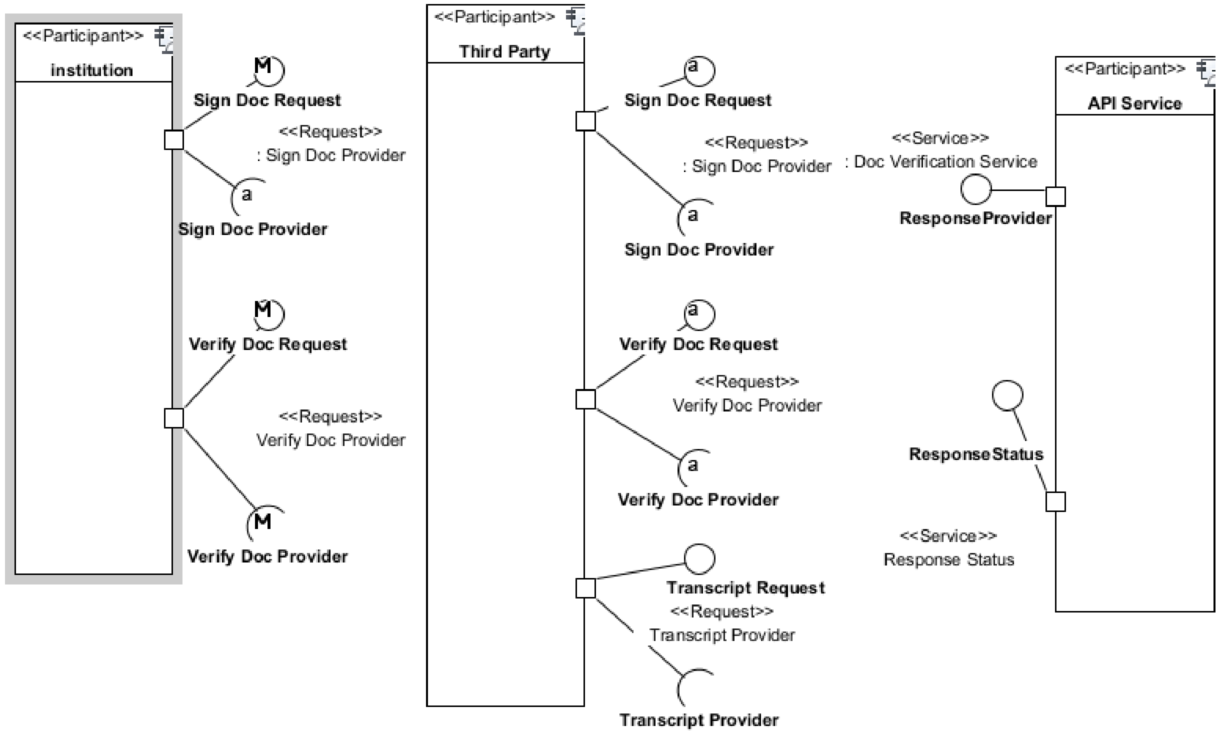


Fig 3.3 Service Participant Diagram for DocVerify

3.7.4 Service Contract Diagram

Service contract defines the agreement between parties about how a service is to be provided and consumed. "Agreement" here refers to interfaces, choreography and any terms and conditions. Interacting participants MUST agree to the agreement in order for the service to be enacted.

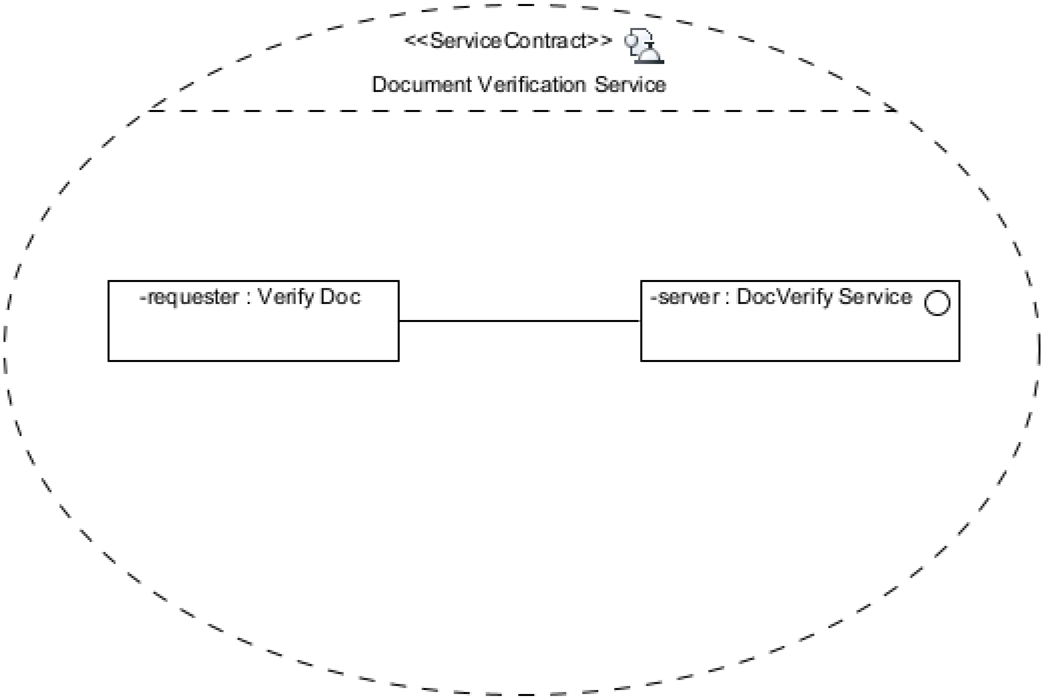


Fig 3.4 Service Contract Diagram for DocVerify

3.7.5 Service Architecture Diagram

Understanding how people, team and organizations work together for a goal enables them to work more cohesively using services without getting overly coupled. SoaML enables modelers to build a services architecture model for this purpose. The services architecture put together the service specification and participants, and shows how they work together to achieve a goal. Services architecture diagram is a SoaML diagram that represents services architecture.

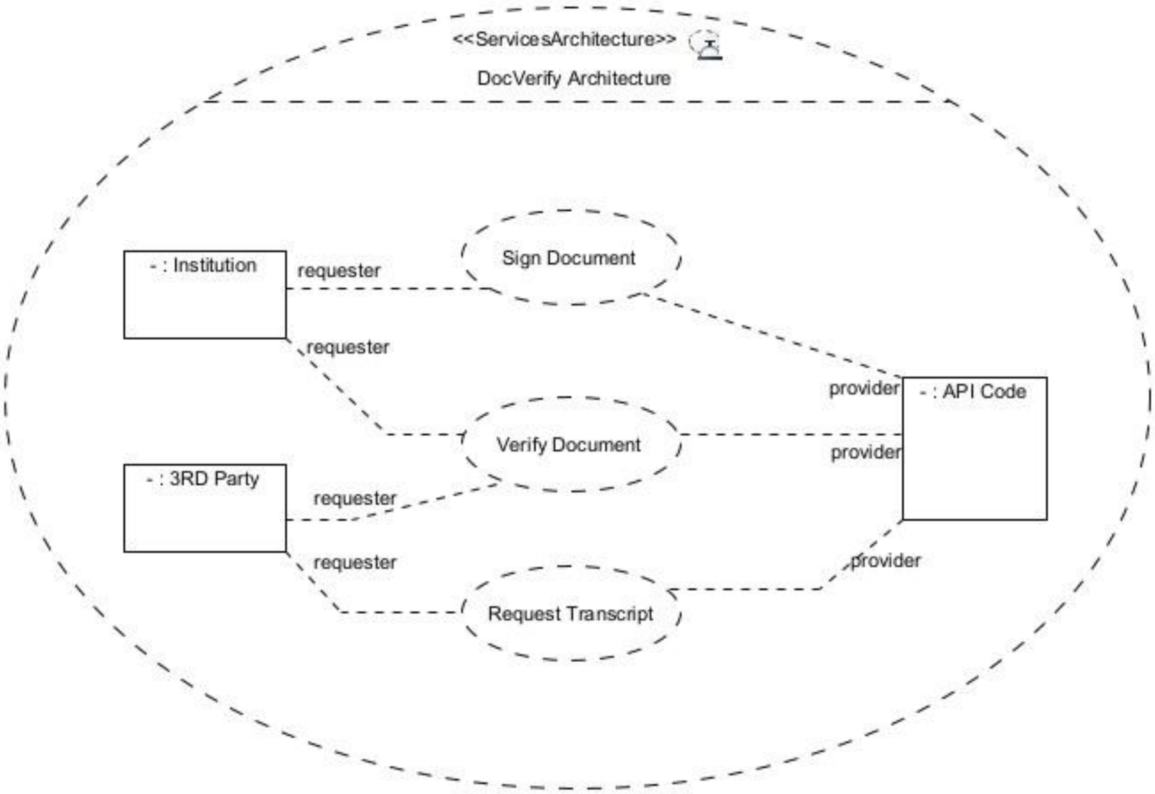


Fig 3.5 Service Architecture Diagram for DocVerify Service

CHAPTER FOUR

IMPLEMENTATION AND DOCUMENTATION

System Implementation is the development, integration and testing of system components and delivery of that system into production. The chapter provides the user guide of the implementation of the system developed and its corresponding documentation.

4.1 Software Development Tools/Technologies

These are the languages, tools, and technologies that has helped in the creation and/or modification and testing of the services implemented in this project.

4.1.1 Implementation Languages

These are the programming languages used to achieve the goal of this project.

4.1.1.1 Python

Python being an interpreted high-level general-purpose programming language, its design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help write clear, logical code for small and large-scale projects.

The selection of this language was based on the familiarity with the language and the framework which enhances the cost of implementation. Further reasons for selection of this language was based on its high readable and dense syntax, developers can express a concept with more ease than they can, using other languages.

4.1.1.1 Flask

Flask is a lightweight ‘web application framework’ written in Python. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to work with. Flask is super useful for building Microservices. It helps you utilize any number of its built-in extensions to design and deploy Microservices at high velocity.

4.1.1.3 HTML, CSS and JavaScript

These trio are the fundamentals of web development. HTML provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript. CSS is used to control formatting, and layout. JavaScript is used to control the behavior of different elements, making HTML pages more dynamic and interactive. These trio has been used together for the frontend implementation of DocVerify.

4.1.2 Software IDE and Tools Used

4.1.2.1 Postman

Postman is an application for testing APIs, by sending request to the web server and getting the response back. It allows users to set up all the headers and cookies the API expects, and checks the response.

A test in Postman is fundamentally a JavaScript code, which run after a request is sent and a response has been received from the server. Postman allows you to easily run UI-driven API tests. As Postman is a powerful HTTP client, RESTful API exploration becomes a breeze. Users can quickly put together simple and complex HTTP requests to test, develop and document APIs in no time.

4.1.2.2 PyCharm IDE

PyCharm is a dedicated Integrated Development Environment (IDE) for Python providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python web, data science, machine language and other Python application areas. It has rich professional support for Flask which is the framework used for this project to build a microservice oriented application.

4.2 Cloud Computing Tools/Technologies

These are the several technologies, tools and providers involved in the continuous integration and delivery of the services of DocVerify.

4.2.1 Docker

Docker is an open platform for developing, shipping, and running applications. You can develop applications very fast and deploy them fast. Using Docker, it is easy to create required services separately and manage them as microservices without affecting other services.

Being a tool designed to make it easier to create, deploy, and run applications by using containers, the containers allows me to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package. By doing so, thanks to the container, I am rest assured that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code.

The microservices built for this project fit nicely with the Docker paradigm. With just simple Dockerfile configuration, the whole microservices are packaged into containers which are all independent and loosely coupled.

4.2.2 Kubernetes

Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable. It allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more. You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container. It restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

4.2.3 Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2’s simple web service interface allows you to obtain and configure virtual machines with minimal effort. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment.

Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model. We offer the fastest processors in the cloud and we are the only cloud with 400 Gbps Ethernet networking. We have the most powerful GPU instances for machine learning training and graphics workloads, as well as the lowest cost-per-inference instances in the cloud.

4.2.4 Amazon RDS (Relational Database) for PostgreSQL

PostgreSQL has become the preferred open source relational database for many enterprise developers and start-ups, powering leading business and mobile applications. Amazon RDS makes it easy to set up, operate, and scale PostgreSQL deployments in the cloud. With Amazon RDS, you can deploy scalable PostgreSQL deployments in minutes with cost-efficient and resizable hardware capacity. Amazon RDS manages complex and time-consuming administrative tasks such as PostgreSQL software installation and upgrades; storage management; replication for high availability and read throughput; and backups for disaster recovery.

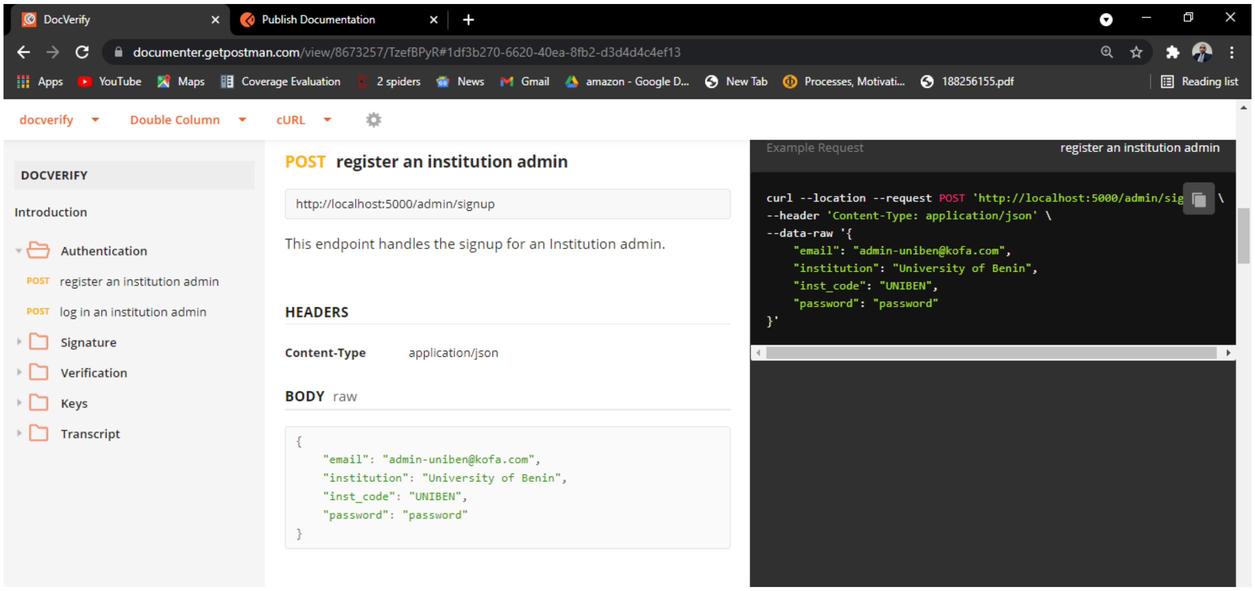
Amazon RDS for PostgreSQL gives you access to the capabilities of the familiar PostgreSQL database engine. This means that the code, applications, and tools you already use today with your existing databases can be used with Amazon RDS.

4.3 Service API Documentation

4.3.1 Authentication Service

The Authentication Service handles all the mechanism that concerns with secure authentication of the institution admin users to the system, involving the registration, validating login to prove identity and also offer authorization tokens to access protected endpoints. This Service contains two endpoints: Signup and Login.

4.3.1.1 /signup



POST register an institution admin

http://localhost:5000/admin/signup

This endpoint handles the signup for an Institution admin.

HEADERS

Content-Type application/json

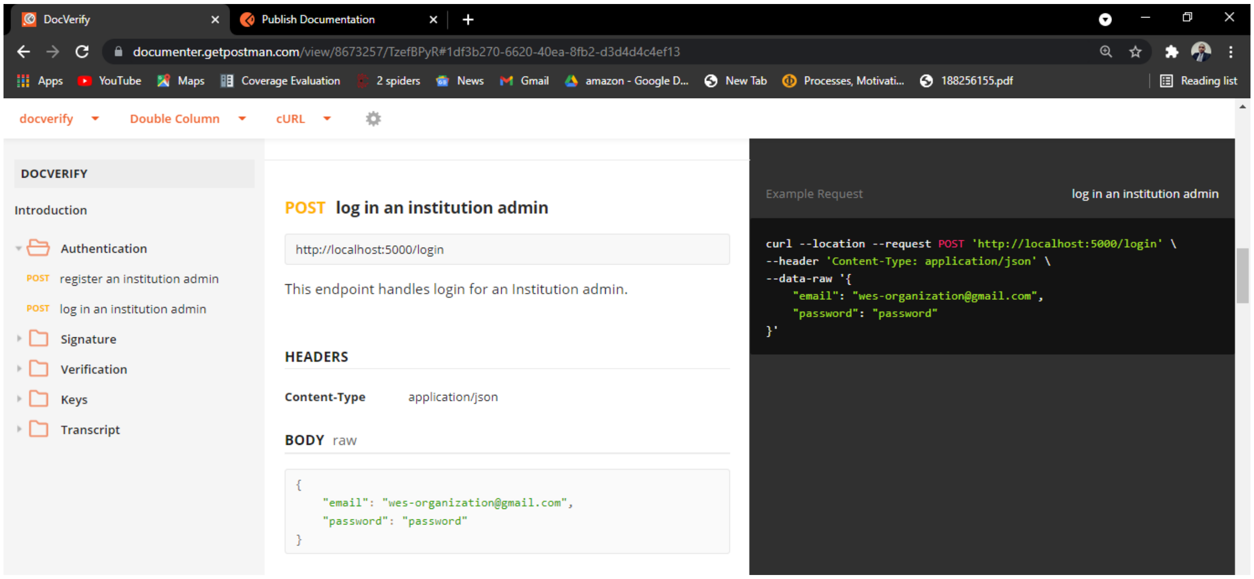
BODY raw

```
{
    "email": "admin-uniben@kofa.com",
    "institution": "University of Benin",
    "inst_code": "UNIBEN",
    "password": "password"
}
```

```
curl --location --request POST 'http://localhost:5000/admin/sig \
--header 'Content-Type: application/json' \
--data-raw '{
    "email": "admin-uniben@kofa.com",
    "institution": "University of Benin",
    "inst_code": "UNIBEN",
    "password": "password"
}'
```

Fig 4.1 DocVerify Signup Endpoint Documentation shot from Postman

4.3.1.2 /login



POST log in an institution admin

http://localhost:5000/login

This endpoint handles login for an Institution admin.

HEADERS

Content-Type	application/json

BODY raw

```
{
    "email": "wes-organization@gmail.com",
    "password": "password"
}
```

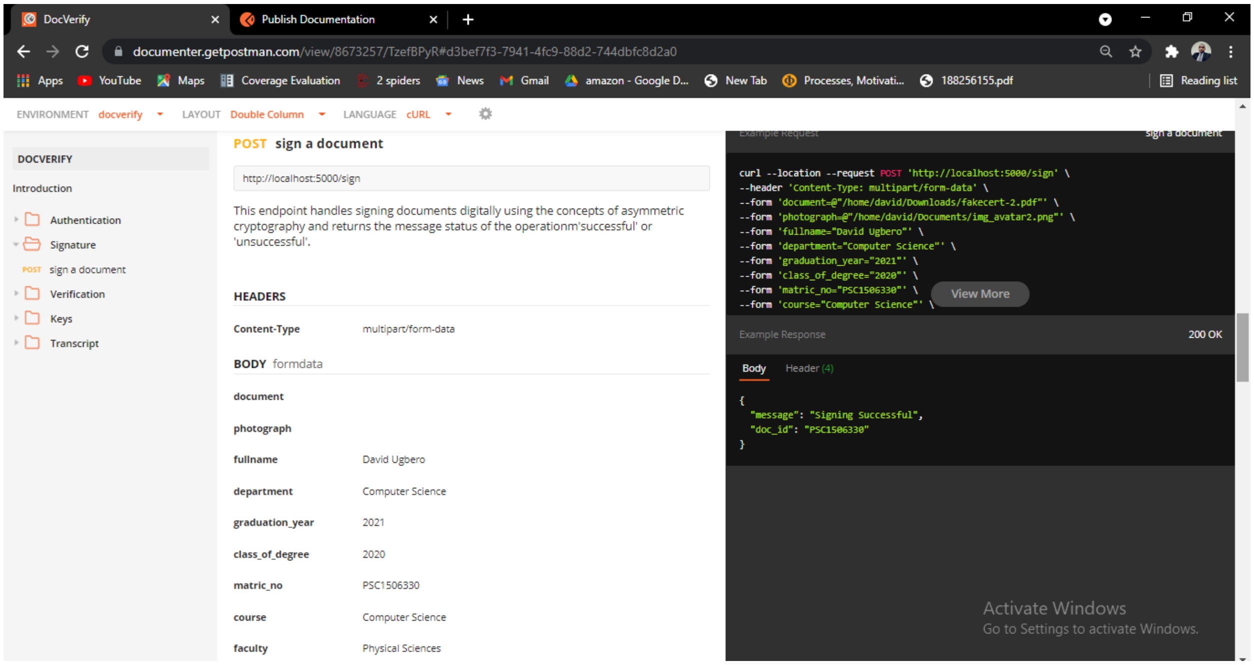
```
curl --location --request POST 'http://localhost:5000/login' \
--header 'Content-Type: application/json' \
--data-raw '{
    "email": "wes-organization@gmail.com",
    "password": "password"
}'
```

Fig 4.2 DocVerify Login Endpoint Documentation shot from Postman

4.3.2 Signature Service

The Signature service covers for the cryptographic signing of documents which encrypts them using the RSA algorithm. It contains one endpoint: Sign.

4.3.2.1 /sign



POST sign a document

http://localhost:5000/sign

This endpoint handles signing documents digitally using the concepts of asymmetric cryptography and returns the message status of the operationm'successful' or 'unsuccessful'.

HEADERS

Content-Type	multipart/form-data

BODY formdata

document	
photograph	
fullname	David Ugbero
department	Computer Science
graduation_year	2021
class_of_degree	2020
matric_no	PSC1506330
course	Computer Science
faculty	Physical Sciences

```
curl --location --request POST 'http://localhost:5000/sign' \
--header 'Content-Type: multipart/form-data' \
--form 'document=@"/home/david/Downloads/fakecert-2.pdf"' \
--form 'photograph=@"/home/david/Documents/img_avatar2.png"' \
--form 'fullname="David Ugbero"' \
--form 'department="Computer Science"' \
--form 'graduation_year="2021"' \
--form 'class_of_degree="2020"' \
--form 'matric_no="PSC1506330"' \
--form 'course="Computer Science"' \
```

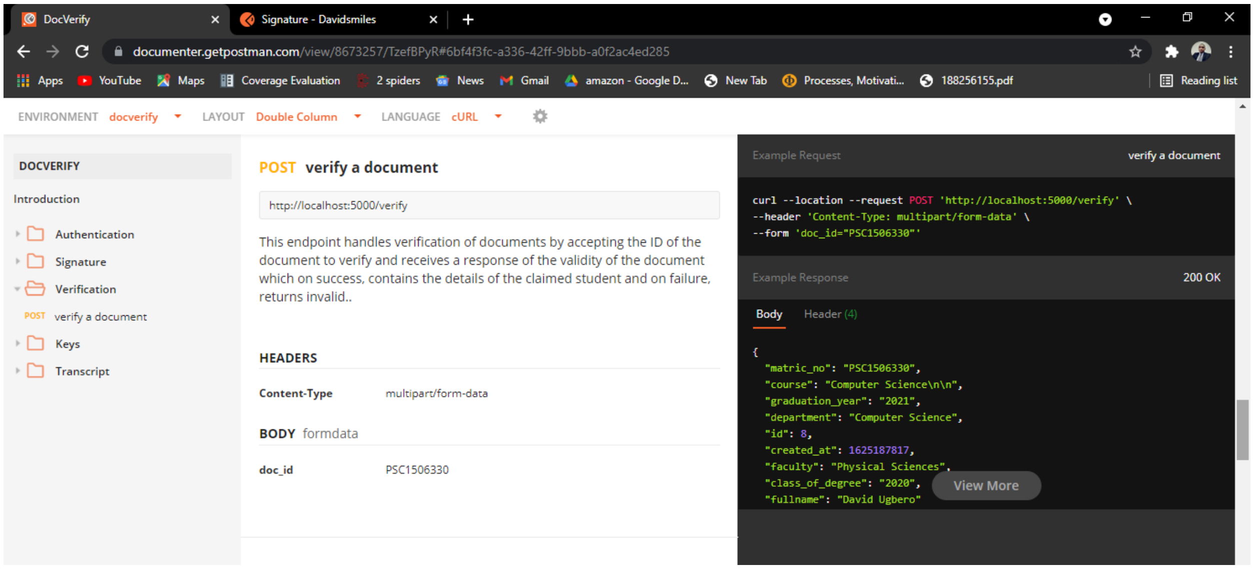
```
{
  "message": "Signing Successful",
  "doc_id": "PSC1506330"
}
```

Fig 4.3 DocVerify Sign Endpoint Documentation shot from Postman

4.3.3 Verification Service

The Verification service handles the cryptography verification of documents that have been digitally signed. It has one endpoint: Verify.

4.3.3.1 /verify



ENVIRONMENT docverify LAYOUT Double Column LANGUAGE cURL

POST verify a document

http://localhost:5000/verify

This endpoint handles verification of documents by accepting the ID of the document to verify and receives a response of the validity of the document which on success, contains the details of the claimed student and on failure, returns invalid..

HEADERS

Content-Type multipart/form-data

BODY formdata

doc_id PSC1506330

Example Request — verify a document

```
curl --location --request POST 'http://localhost:5000/verify' \
--header 'Content-Type: multipart/form-data' \
--form 'doc_id="PSC1506330"'
```

Example Response — 200 OK

Body Header (4)

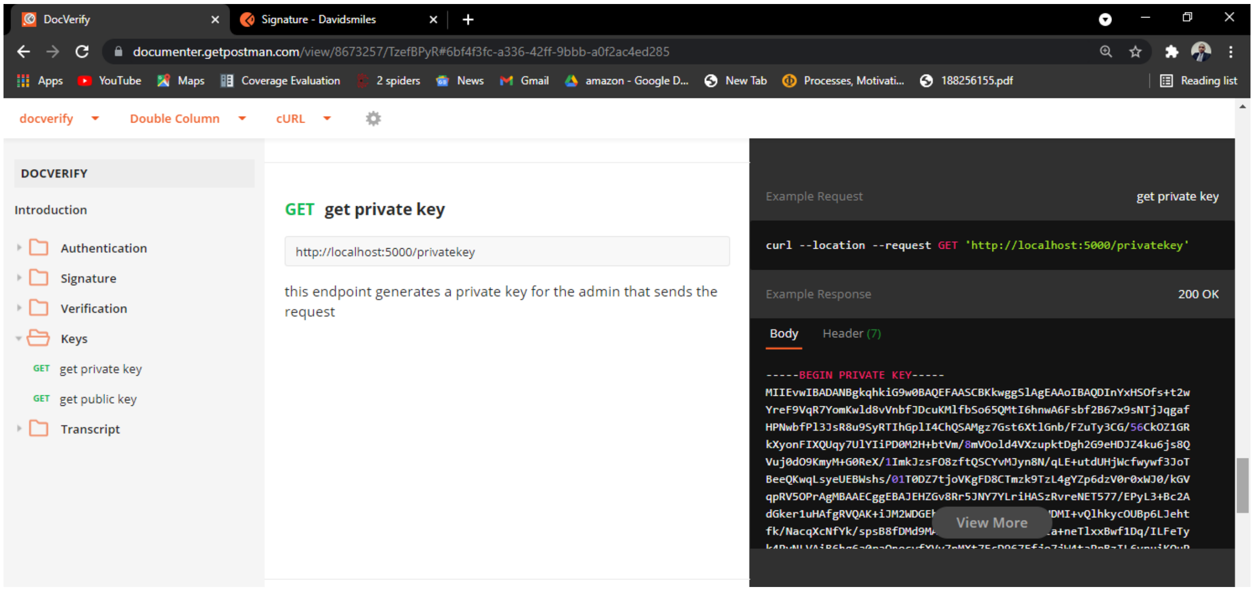
```
{
  "matric_no": "PSC1506330",
  "course": "Computer Science\n\n",
  "graduation_year": "2021",
  "department": "Computer Science",
  "id": 8,
  "created_at": 1625187817,
  "faculty": "Physical Sciences",
  "class_of_degree": "2020",
  "fullname": "David Ugbero"
```

Fig 4.4 DocVerify Verify Endpoint Documentation shot from Postman

4.3.4 Keys Service

This Keys service is responsible for handling and managing the public and private keys used by the users involved in this system. It contains two endpoints: Private Key and Public Key

4.3.4.1 /privatekey



GET get private key

http://localhost:5000/privatekey

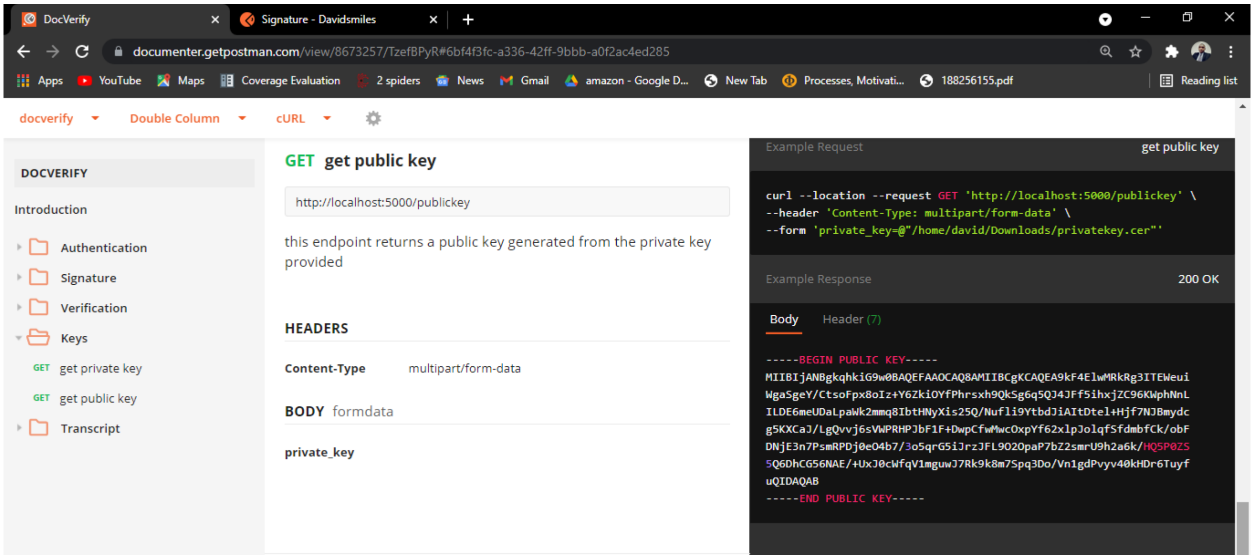
this endpoint generates a private key for the admin that sends the request

```
curl --location --request GET 'http://localhost:5000/privatekey'
```

```
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDInYxHSOfs+t2w
YreF9VqR7YomKwld8vVnbfJDcuKMlfbSo65QMtI6hnwA6Fsbf2B67x9sNTjJqgaf
HPNwbfPl3JsR8u9SyRTIhGplI4ChQSAMgz7Gst6XtlGnb/FZuTy3CG/56CkOZ1GR
kXyonFIXQUqy7UlYIiPD0M2H+btVm/8mVOold4VXzupktDgh2G9eHDJZ4ku6js8Q
Vuj0dO9KmyM+G0ReX/1ImkJzsFO8zftQSCYvMJyn8N/qLE+utdUHjWcfwywf3JoT
BeeQKwqLsyeUEBWshs/01T0DZ7tjoVKgFD8CTmzk9TzL4gYZp6dzV0r0xWJ0/kGV
qpRV5OPrAgMBAAECggEBAJEHZGv8Rr5JNY7YLriHASzRvreNET577/EPyL3+Bc2A
dGker1uHAfgRVQAK+iJM2WDGEh          DMI+vQlhkycOUBp6LJeht
fk/NacqXcNfYk/spsB8fDMd9MA          a+neTlxxBwf1Dq/ILFeTy
```

Fig 4.5 DocVerify PrivateKey Endpoint Documentation shot from Postman

4.3.4.2 /publickey



GET get public key

http://localhost:5000/publickey

this endpoint returns a public key generated from the private key provided

HEADERS

Content-Type	multipart/form-data

BODY formdata

private_key

```
curl --location --request GET 'http://localhost:5000/publickey' \
--header 'Content-Type: multipart/form-data' \
--form 'private_key=@"/home/david/Downloads/privatekey.cer"'
```

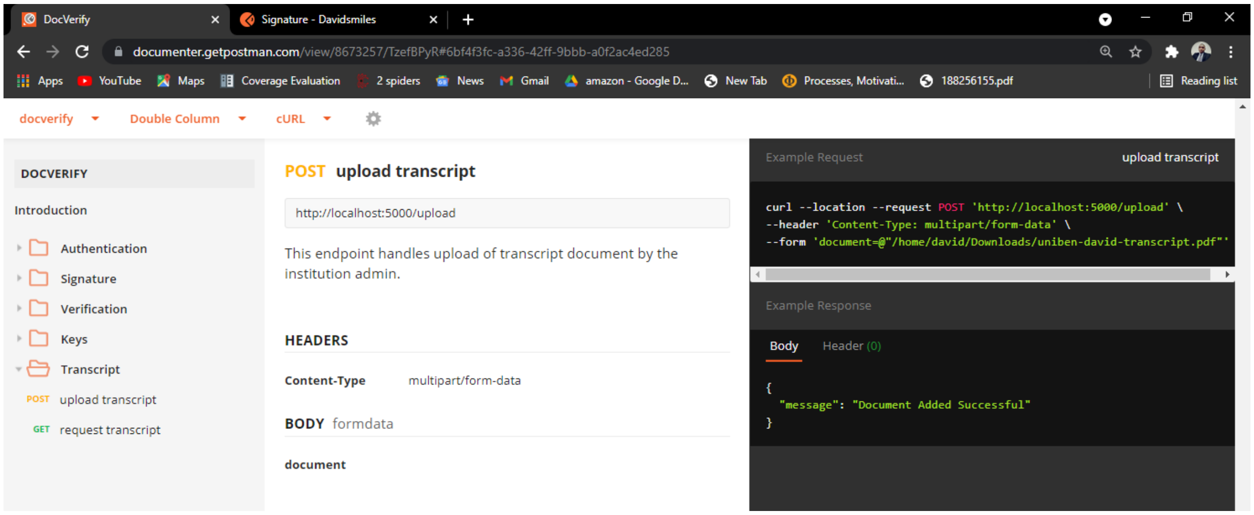
```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA9kF4ElwMRkRg3ITEWeui
WgaSgeY/CtsoFpx8oIz+Y6ZkiOYfPhrsxh9QkSg6q5QJ4JFf5ihxjZC96KWphNnL
ILDE6meUDaLpaWk2mmq8IbtHNyXis25Q/Nufli9YtbdJiAItDtel+Hjf7NJBmydc
g5KXCaJ/LgQvvj6sVWPRHPJbF1F+DwpCfwMwcOxpYf62xlpJolqfSfdmbfCk/obF
DNjE3n7PsmRPDj0eO4b7/3o5qrG5iJrzJFL9O2OpaP7bZ2smrU9h2a6k/HQ5P0ZS
5Q6DhCG56NAE/+UxJ0cWfqV1mguwJ7Rk9k8m7Spq3Do/Vn1gdPvyv40kHDr6Tuyf
uQIDAQAB
-----END PUBLIC KEY-----
```

Fig 4.6 DocVerify PublicKey Endpoint Documentation shot from Postman

4.3.5 Transcript Service

The Transcript Service handles the upload and request of transcript document by the Institution Admin and Third Party system respectively. It has two endpoints: Upload ad Request Transcript.

4.3.5.1 /upload



POST upload transcript

http://localhost:5000/upload

This endpoint handles upload of transcript document by the institution admin.

HEADERS

Content-Type multipart/form-data

BODY formdata

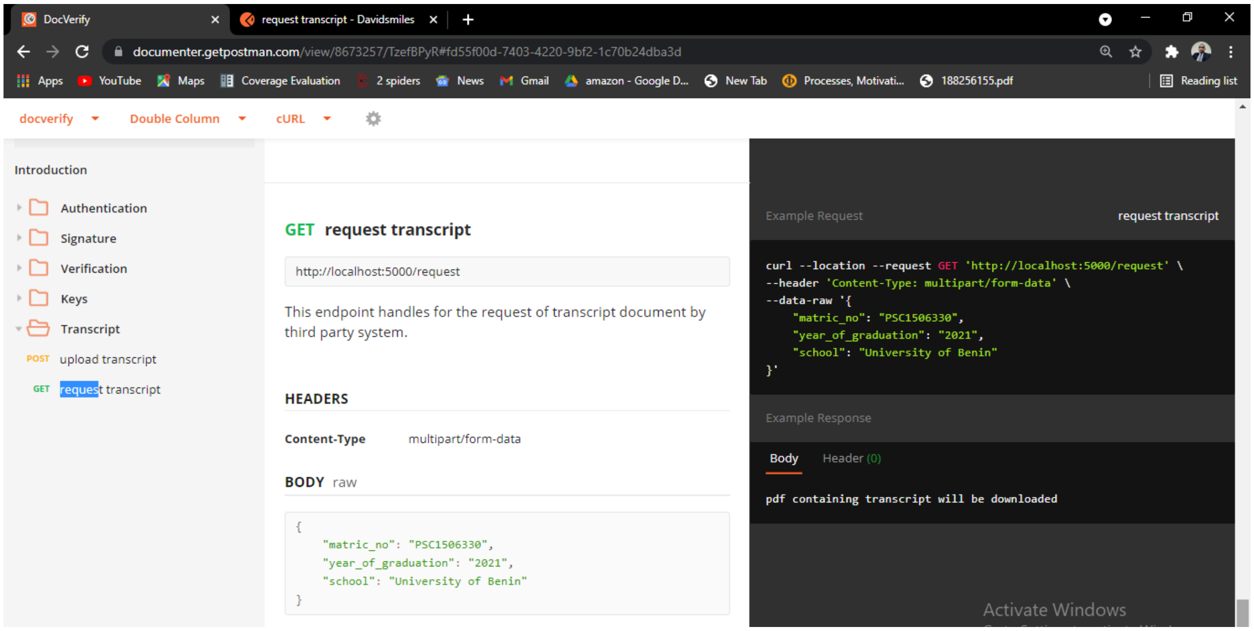
document

```
curl --location --request POST 'http://localhost:5000/upload' \
--header 'Content-Type: multipart/form-data' \
--form 'document=@"/home/david/Downloads/uniben-david-transcript.pdf"'
```

```
{
  "message": "Document Added Successful"
}
```

Fig 4.7 DocVerify Upload Endpoint Documentation shot from Postman

4.3.5.2 /request



GET request transcript

http://localhost:5000/request

This endpoint handles for the request of transcript document by third party system.

HEADERS

Content-Type multipart/form-data

BODY raw

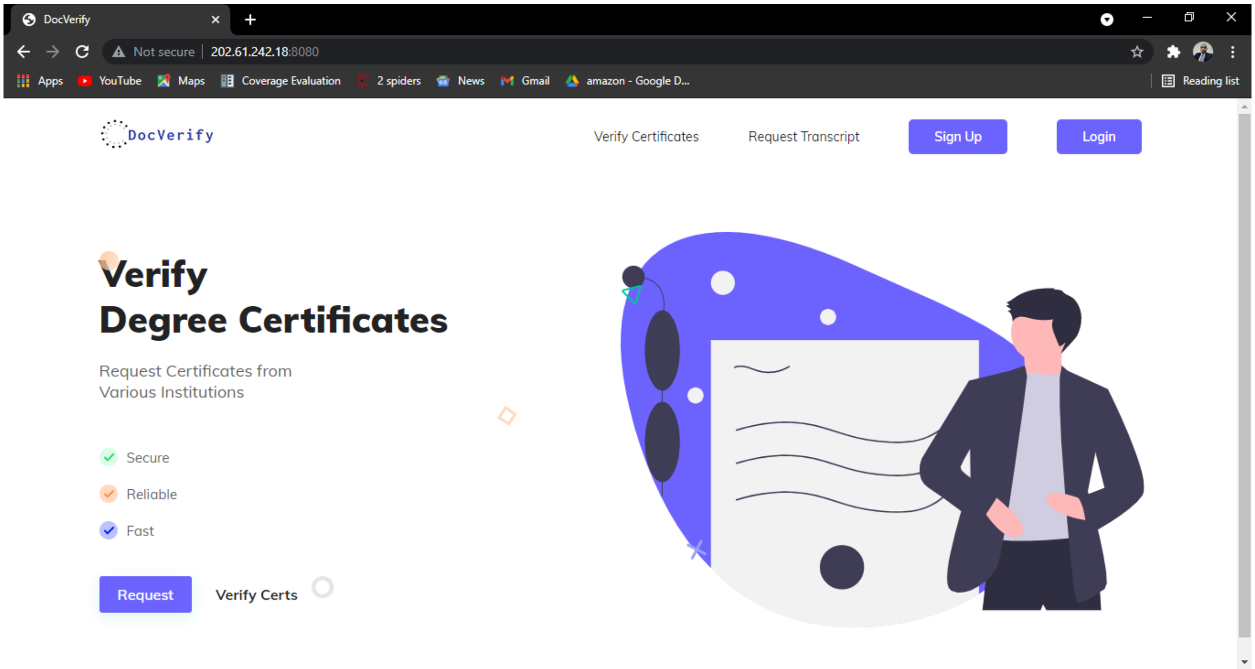
```
{
    "matric_no": "PSC1506330",
    "year_of_graduation": "2021",
    "school": "University of Benin"
}
```

```
curl --location --request GET 'http://localhost:5000/request' \
--header 'Content-Type: multipart/form-data' \
--data-raw '{
    "matric_no": "PSC1506330",
    "year_of_graduation": "2021",
    "school": "University of Benin"
}'
```

```
pdf containing transcript will be downloaded
```

Fig 4.8 DocVerify Request Transcript Endpoint Documentation shot from Postman

4.4 User Guide



Verify
Degree Certificates

Request Certificates from
Various Institutions

Fig 4.9 DocVerify HomePage Screenshot

4.4.1 Home Page

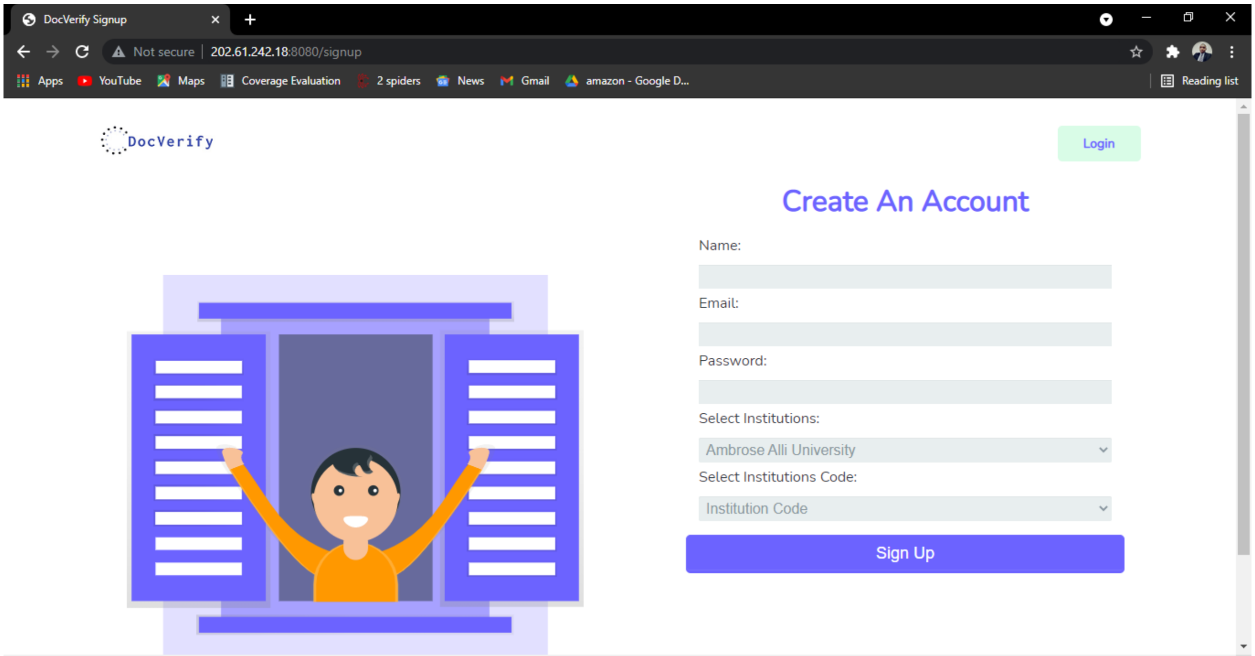
This is a screenshot of the Home Page of DocVerify, this page contains the various buttons to navigate through all the pages in the web application and also a picture that describe what the system does.

The System is free and accessible to all Third Party users (individuals, companies, educational institutions) that wishes to verify a particular certificate document.

However, the system provides a Sign Up and Login Page for an Institution Admin to carry out administrative tasks on DocVerify.

Both Sign Up and Login pages are shown below:

4.4.2 Sign Up and Login Page



Create An Account

Name:

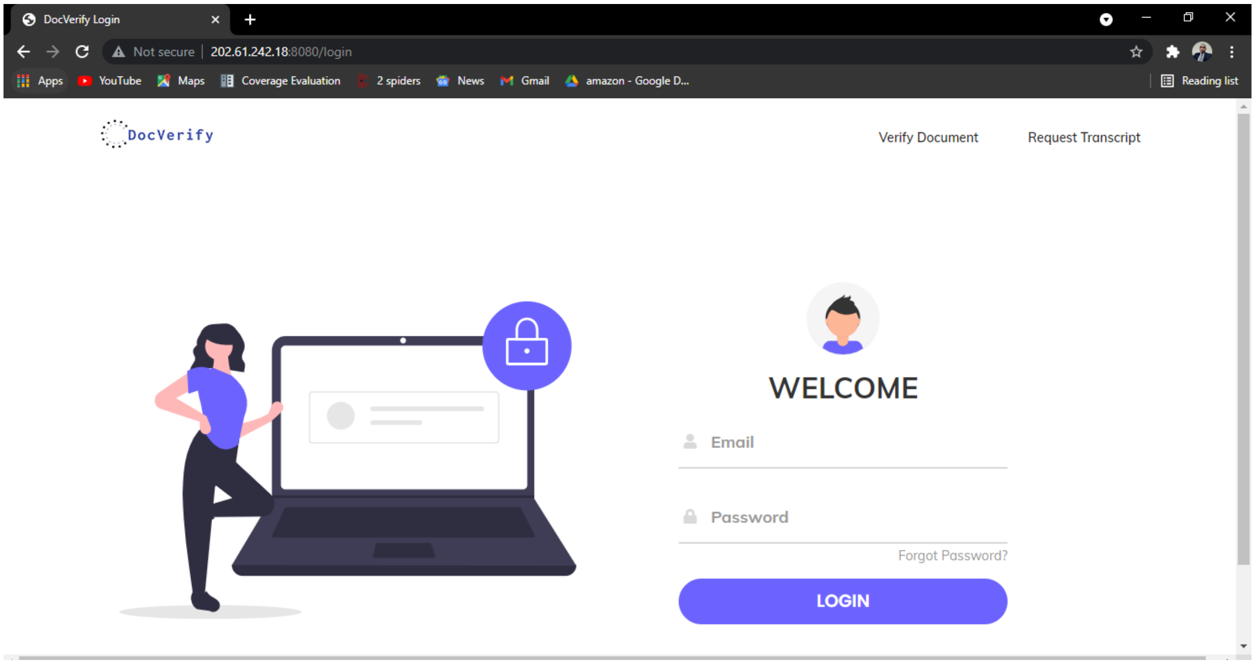
Email:

Password:

Select Institutions: Ambrose Alli University

Select Institutions Code: Institution Code

Fig 4.10 DocVerify SignUp Screenshot



WELCOME

Email

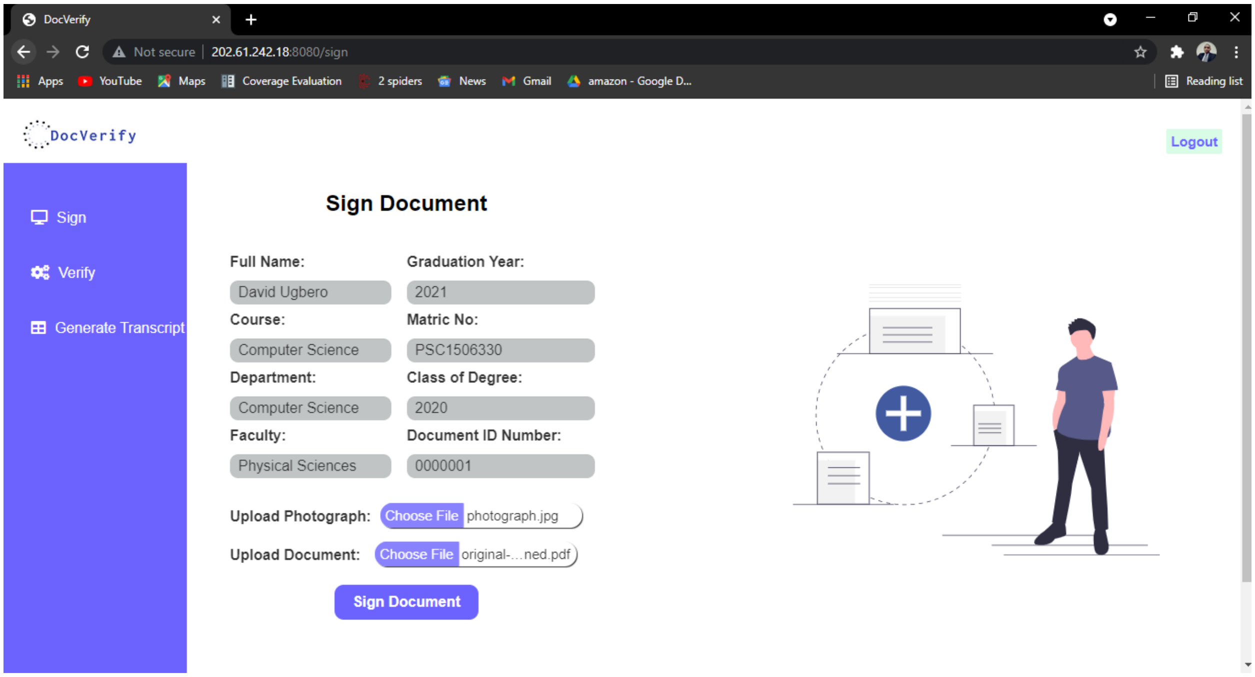
Password

Fig 4.11 DocVerify Login Screenshot

Upon successful login, the Institution Admin is presented with a dashboard that contains functions he can perform on the system.

This Dashboard contains three (3) services or features that the Admin can perform: Sign a Document, Verify and Generate Transcript.

4.4.3 Sign a Document Page



Sign Document

Full Name: David Ugbero

Graduation Year: 2021

Course: Computer Science

Matric No: PSC1506330

Department: Computer Science

Class of Degree: 2020

Faculty: Physical Sciences

Document ID Number: 0000001

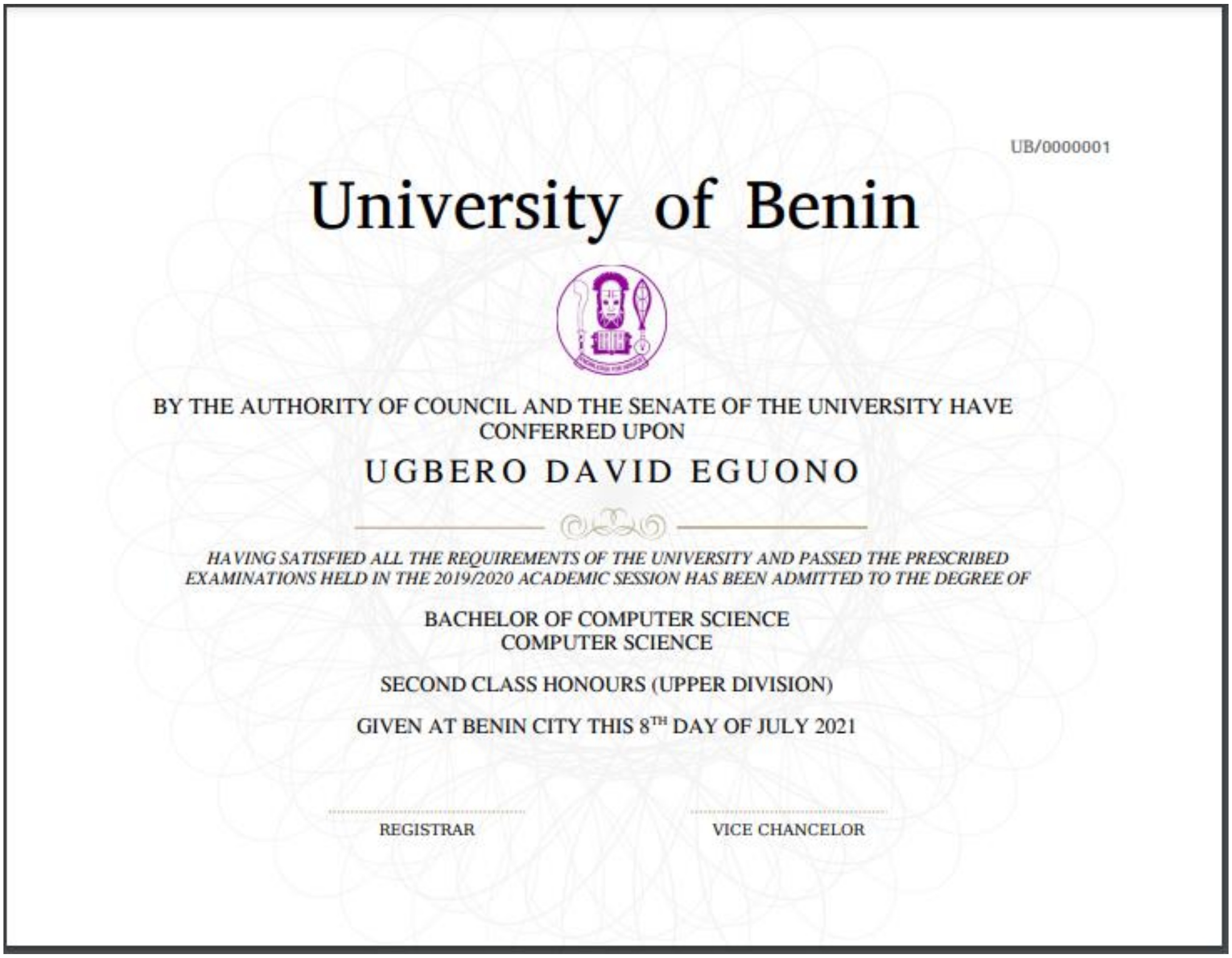
Upload Photograph: photograph.jpg

Upload Document: original-...ned.pdf

Fig 4.12 DocVerify Sign a Document Page Screenshot

This page is available to the institution admin, then the Signature Service code accepts all the parameters provided here to digitally sign the document using asymmetric cryptography with RSA algorithm.

The following screenshots shows how the original certificate document looks like before being signed.



UB/0000001

University of Benin

BY THE AUTHORITY OF COUNCIL AND THE SENATE OF THE UNIVERSITY HAVE CONFERRED UPON

UGBERO DAVID EGUONO

HAVING SATISFIED ALL THE REQUIREMENTS OF THE UNIVERSITY AND PASSED THE PRESCRIBED EXAMINATIONS HELD IN THE 2019/2020 ACADEMIC SESSION HAS BEEN ADMITTED TO THE DEGREE OF

BACHELOR OF COMPUTER SCIENCE
COMPUTER SCIENCE

SECOND CLASS HONOURS (UPPER DIVISION)

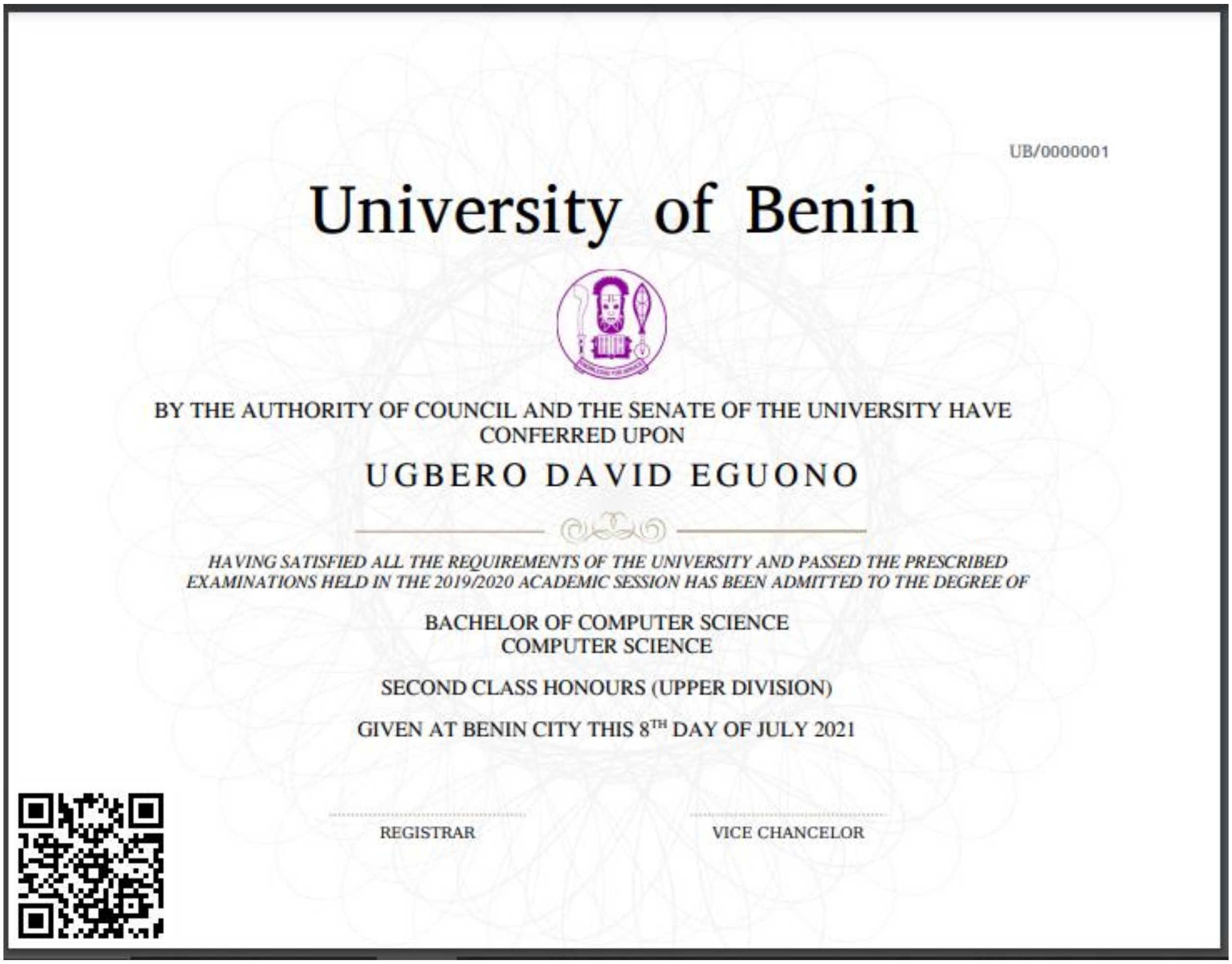
GIVEN AT BENIN CITY THIS 8TH DAY OF JULY 2021

REGISTRAR

VICE CHANCELOR

Fig 4.13 Original Certificate Before Signed

Once the admin clicks on the Sign Document button on the /sign page, the Signature Service returns a signed document of the certificate which is then downloaded to the admin's system or opened as shown below.



UB/0000001

University of Benin

BY THE AUTHORITY OF COUNCIL AND THE SENATE OF THE UNIVERSITY HAVE CONFERRED UPON

UGBERO DAVID EGUONO

HAVING SATISFIED ALL THE REQUIREMENTS OF THE UNIVERSITY AND PASSED THE PRESCRIBED EXAMINATIONS HELD IN THE 2019/2020 ACADEMIC SESSION HAS BEEN ADMITTED TO THE DEGREE OF

BACHELOR OF COMPUTER SCIENCE
COMPUTER SCIENCE

SECOND CLASS HONOURS (UPPER DIVISION)

GIVEN AT BENIN CITY THIS 8TH DAY OF JULY 2021

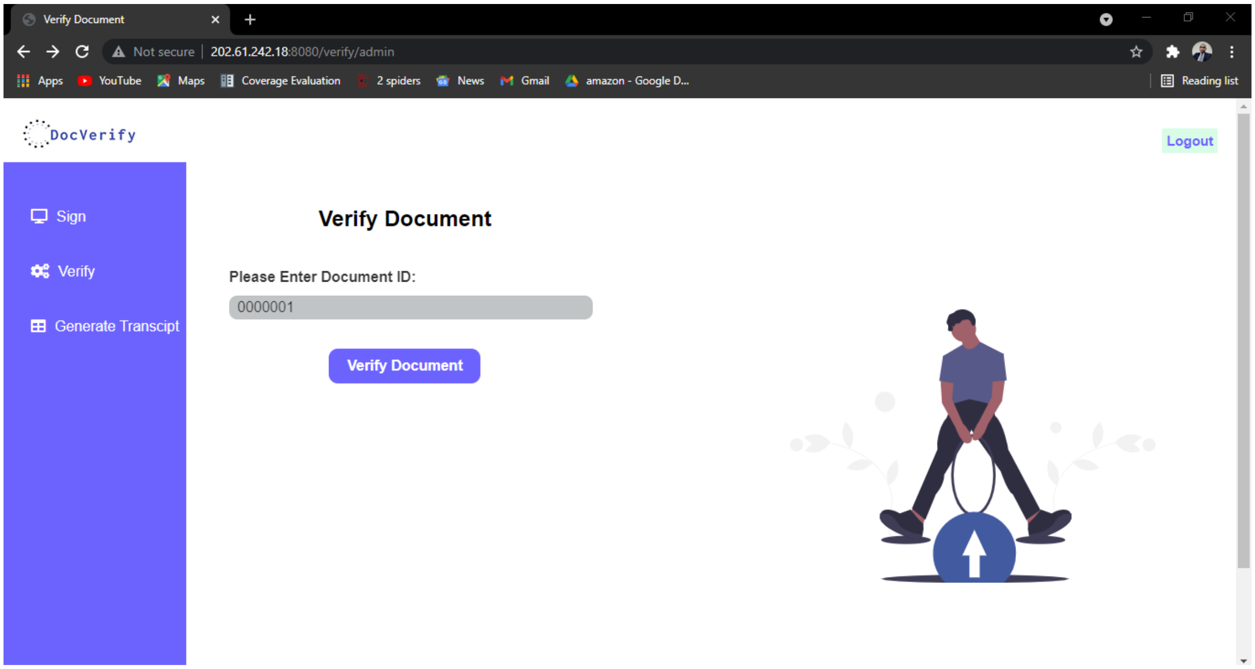
REGISTRAR

VICE CHANCELOR

Fig 4.14 Output of Certificate after Signed

4.4.4 Verify Page

The following screenshot here shows the Verify Page which is available to both the Institution Admin and Third party systems like individuals, companies or WES to verify the claims of a certificate document.


Verify Document
Not secure | 202.61.242.18:8080/verify/admin
Apps YouTube Maps Coverage Evaluation 2 spiders News Gmail amazon - Google D...
Reading list
DocVerify
Logout
Sign
Verify
Generate Transcipt

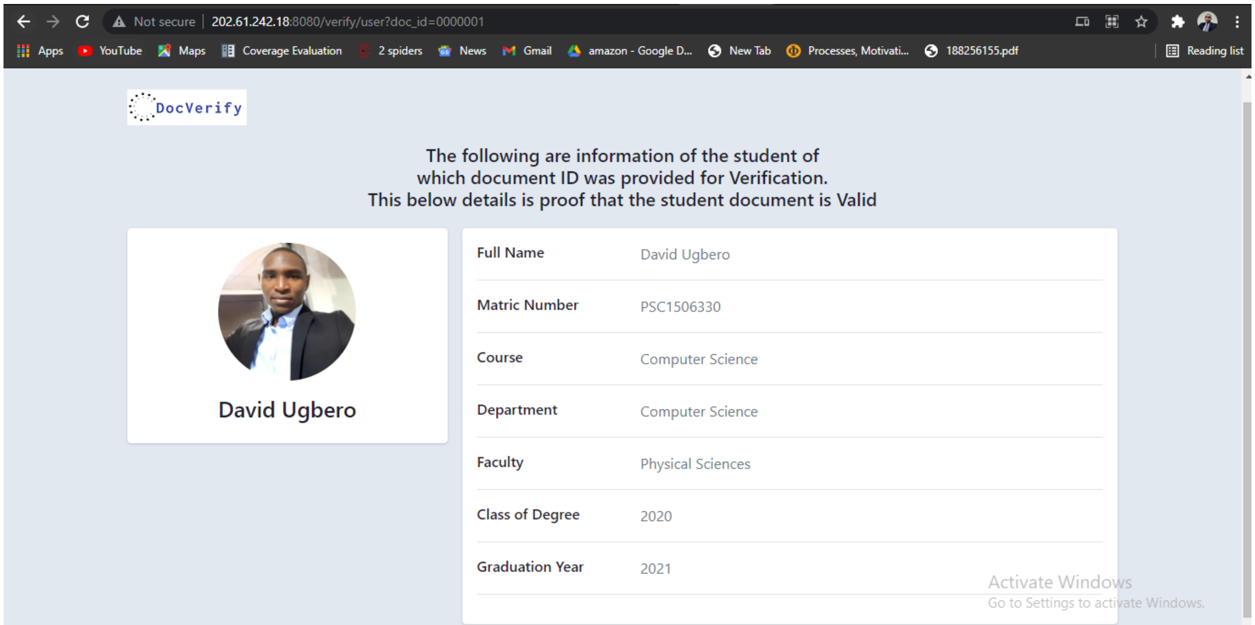
Verify Document

Please Enter Document ID:

0000001

Verify Document

Fig 4.15 DocVerify Verify Page Screenshot

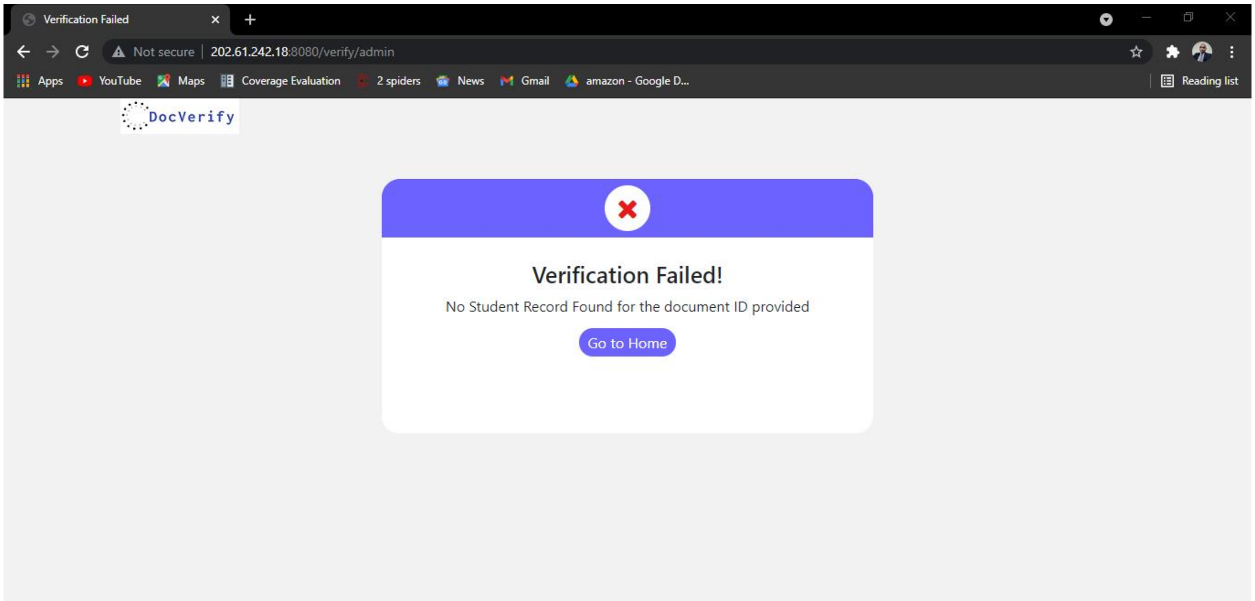

Not secure | 202.61.242.18:8080/verify/user?doc_id=0000001
Apps YouTube Maps Coverage Evaluation 2 spiders News Gmail amazon - Google D... New Tab Processes, Motivati... 188256155.pdf
Reading list
DocVerify
Activate Windows
Go to Settings to activate Windows.

The following are information of the student of
which document ID was provided for Verification.
This below details is proof that the student document is Valid

David Ugbero

Full Name	David Ugbero
Matric Number	PSC1506330
Course	Computer Science
Department	Computer Science
Faculty	Physical Sciences
Class of Degree	2020
Graduation Year	2021

Fig 4.16 DocVerify Profile Page of Student Record on Validation Success Screenshot


Verification Failed
Not secure | 202.61.242.18:8080/verify/admin
Apps YouTube Maps Coverage Evaluation 2 spiders News Gmail amazon - Google D...
Reading list
DocVerify

Verification Failed!

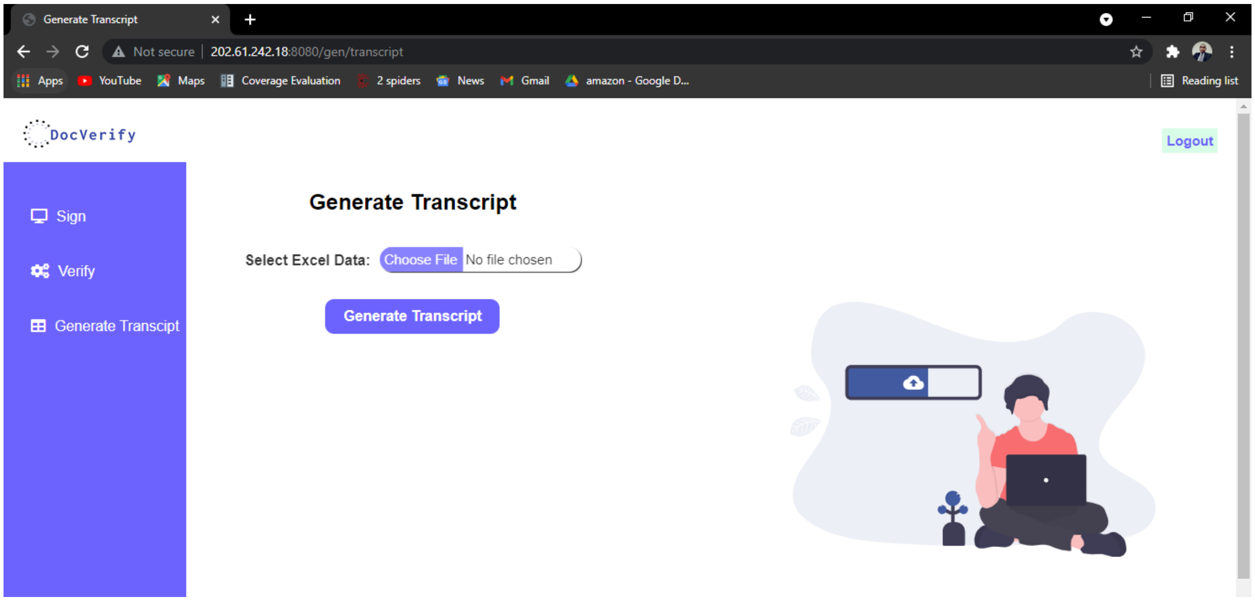
No Student Record Found for the document ID provided

Fig 4.17 DocVerify Validation Failed Page Response Screenshot

4.4.5 Transcript Page

DocVerify adds an extra functionality that allows the Institution Admin to upload Transcript data of Students. This improves on the manual process of Transcript generation in many Nigeria Institutions.

The system also enables for any Third Party Organization to request for transcript data of a student from any Nigeria Institution of choice providing the right parameters.

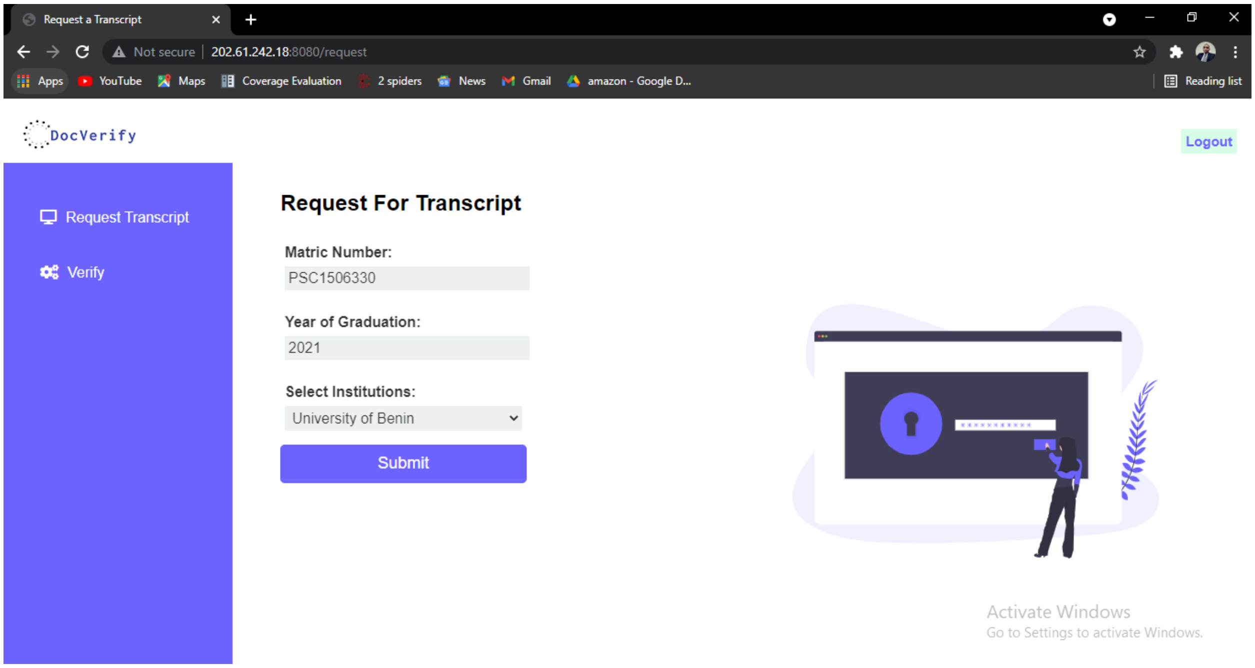

Generate Transcript
Not secure | 202.61.242.18:8080/gen/transcript
Apps YouTube Maps Coverage Evaluation 2 spiders News Gmail amazon - Google D...
Reading list
DocVerify
Logout
Sign
Verify
Generate Transcipt

Generate Transcript

Select Excel Data: Choose File No file chosen

Fig 4.18 DocVerify Generate Transcript Page (for Admins) Screenshot

The Generate Transcript button when clicked, generates a transcript document which gets downloaded into the Admin’s computer and opened in the browser. The image of the generated transcript is shown in **Fig 4.20.**



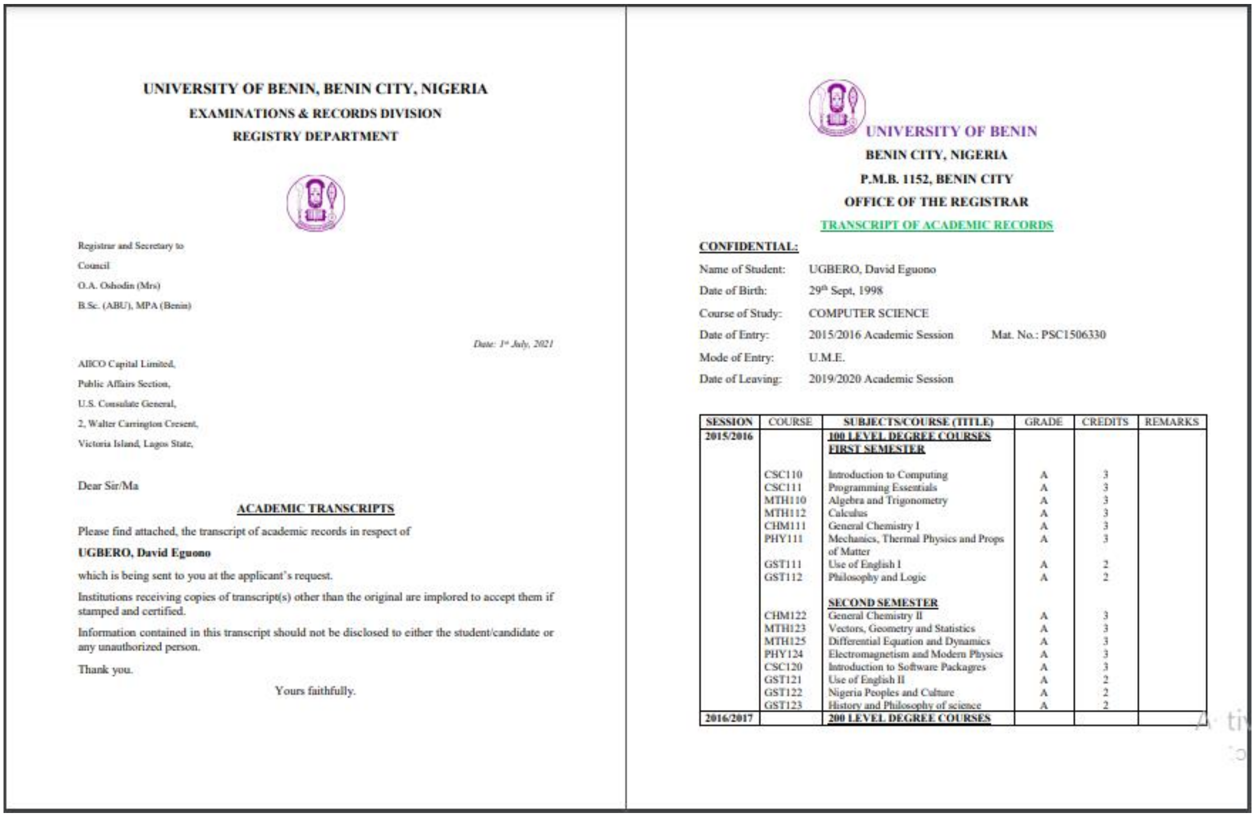
Request For Transcript

Matric Number: PSC1506330

Year of Graduation: 2021

Select Institutions: University of Benin

Fig 4.19 DocVerify Request Transcript Page Screenshot



UNIVERSITY OF BENIN, BENIN CITY, NIGERIA
EXAMINATIONS & RECORDS DIVISION
REGISTRY DEPARTMENT

Registrar and Secretary to
Council
O.A. Oshodin (Mrs)
B.Sc. (ABU), MPA (Benin)

Date: 1st July, 2021

AIICO Capital Limited,
Public Affairs Section,
U.S. Consulate General,
2, Walter Carrington Cresent,
Victoria Island, Lagos State,

Dear Sir/Ma

ACADEMIC TRANSCRIPTS

Please find attached, the transcript of academic records in respect of

UGBERO, David Eguono

which is being sent to you at the applicant's request.

Institutions receiving copies of transcript(s) other than the original are implored to accept them if stamped and certified.

Information contained in this transcript should not be disclosed to either the student/candidate or any unauthorized person.

Thank you.

Yours faithfully.

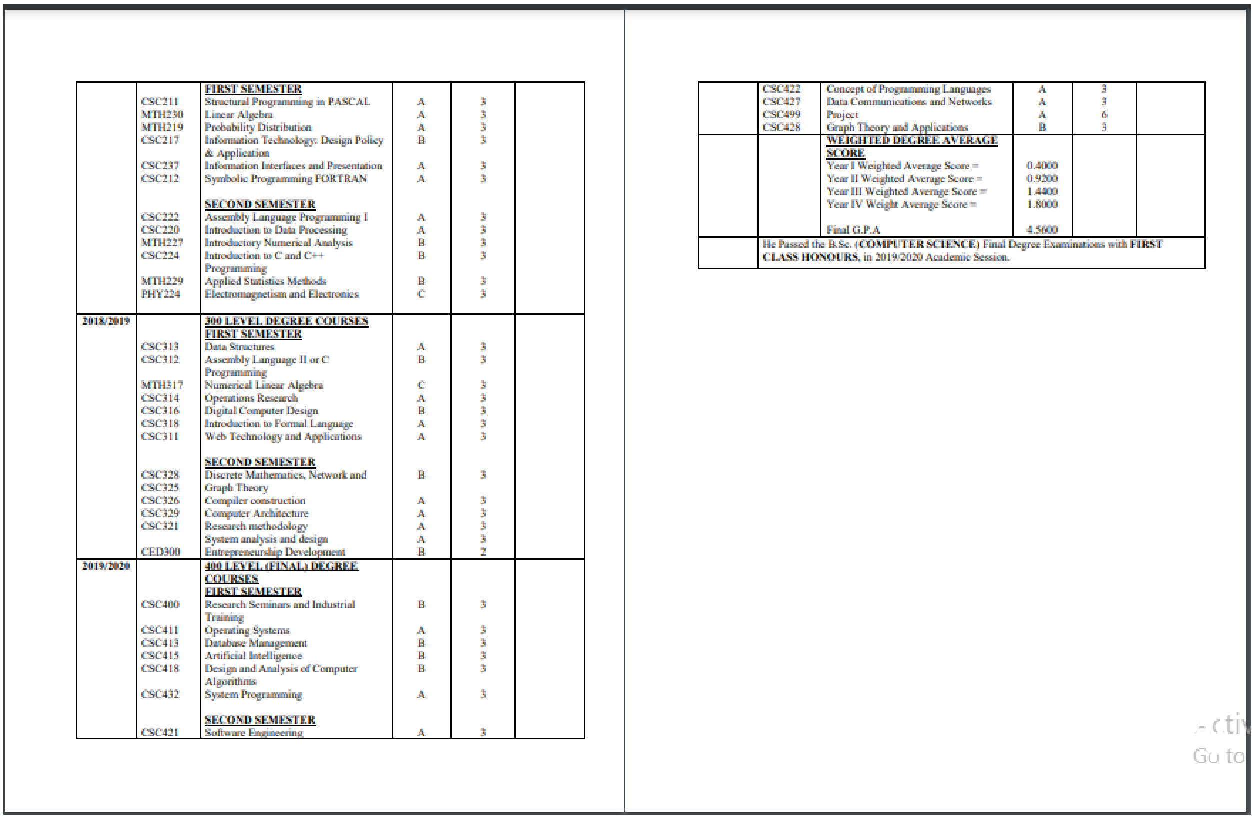
UNIVERSITY OF BENIN
BENIN CITY, NIGERIA
P.M.B. 1152, BENIN CITY
OFFICE OF THE REGISTRAR
TRANSCRIPT OF ACADEMIC RECORDS

CONFIDENTIAL:

Name of Student: UGBERO, David Eguono
Date of Birth: 29th Sept, 1998
Course of Study: COMPUTER SCIENCE
Date of Entry: 2015/2016 Academic Session Mat. No.: PSC1506330
Mode of Entry: U.M.E.
Date of Leaving: 2019/2020 Academic Session

SESSION	COURSE	SUBJECTS/COURSE (TITLE)	GRADE	CREDITS	REMARKS
2015/2016		**100 LEVEL DEGREE COURSES** **FIRST SEMESTER**			
	CSC110	Introduction to Computing	A	3	
	CSC111	Programming Essentials	A	3	
	MTH110	Algebra and Trigonometry	A	3	
	MTH112	Calculus	A	3	
	CHM111	General Chemistry I	A	3	
	PHY111	Mechanics, Thermal Physics and Props of Matter	A	3	
	GST111	Use of English I	A	2	
	GST112	Philosophy and Logic	A	2	
		SECOND SEMESTER			
	CHM122	General Chemistry II	A	3	
	MTH123	Vectors, Geometry and Statistics	A	3	
	MTH125	Differential Equation and Dynamics	A	3	
	PHY124	Electromagnetism and Modern Physics	A	3	
	CSC120	Introduction to Software Packagres	A	3	
	GST121	Use of English II	A	2	
	GST122	Nigeria Peoples and Culture	A	2	
	GST123	History and Philosophy of science	A	2	
2016/2017		**200 LEVEL DEGREE COURSES**			

Fig 4.20 Output of Transcript Pdf File Sent to Requester



		FIRST SEMESTER			
	CSC211	Structural Programming in PASCAL	A	3	
	MTH230	Linear Algebra	A	3	
	MTH219	Probability Distribution	A	3	
	CSC217	Information Technology: Design Policy & Application	B	3	
	CSC237	Information Interfaces and Presentation	A	3	
	CSC212	Symbolic Programming FORTRAN	A	3	
		SECOND SEMESTER			
	CSC222	Assembly Language Programming I	A	3	
	CSC220	Introduction to Data Processing	A	3	
	MTH227	Introductory Numerical Analysis	B	3	
	CSC224	Introduction to C and C++ Programming	B	3	
	MTH229	Applied Statistics Methods	B	3	
	PHY224	Electromagnetism and Electronics	C	3	
2018/2019		**300 LEVEL DEGREE COURSES**			
		FIRST SEMESTER			
	CSC313	Data Structures	A	3	
	CSC312	Assembly Language II or C Programming	B	3	
	MTH317	Numerical Linear Algebra	C	3	
	CSC314	Operations Research	A	3	
	CSC316	Digital Computer Design	B	3	
	CSC318	Introduction to Formal Language	A	3	
	CSC311	Web Technology and Applications	A	3	
		SECOND SEMESTER			
	CSC328	Discrete Mathematics, Network and	B	3	
	CSC325	Graph Theory			
	CSC326	Compiler construction	A	3	
	CSC329	Computer Architecture	A	3	
	CSC321	Research methodology	A	3	
		System analysis and design	A	3	
	CED300	Entrepreneurship Development	B	2	
2019/2020		**400 LEVEL (FINAL) DEGREE COURSES**			
		FIRST SEMESTER			
	CSC400	Research Seminars and Industrial Training	B	3	
	CSC411	Operating Systems	A	3	
	CSC413	Database Management	B	3	
	CSC415	Artificial Intelligence	B	3	
	CSC418	Design and Analysis of Computer Algorithms	B	3	
	CSC432	System Programming	A	3	
		SECOND SEMESTER			
	CSC421	Software Engineering	A	3	
	CSC422	Concept of Programming Languages	A	3	
	CSC427	Data Communications and Networks	A	3	
	CSC499	Project	A	6	
	CSC428	Graph Theory and Applications	B	3	
		WEIGHTED DEGREE AVERAGE SCORE			
		Year I Weighted Average Score =	0.4000		
		Year II Weighted Average Score =	0.9200		
		Year III Weighted Average Score =	1.4400		
		Year IV Weight Average Score =	1.8000		
		Final G.P.A	4.5600		
	He Passed the B.Sc. (**COMPUTER SCIENCE**) Final Degree Examinations with **FIRST CLASS HONOURS**, in 2019/2020 Academic Session.				

Fig 4.21 Output of Transcript Pdf File (Cont'd)

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

With the rise of human population came the rise of educational institution to accommodate the population. However, this has introduced some problems to many organizations and to the educational institutions themselves. One of the most important problems is Verification. There are ways institutes try to combat frauds and forgeries however mostly are time consuming because they are manual and it involves human interaction. The time spent is either reaching out to the university to verify a given document or awaiting a reply from the university that issue the document. The latter can be extremely exhausting and expensive especially if a company does it for several hundreds of applicants.

This research work focuses on curbing these fraudulent documents activities by various means and establishes the fact that a distributed Document Verification Service is one of the most efficient ways of curbing Document Fraud. It began by establishing a goal which is to build a service-based system centered on integrating heterogeneous databases for the purpose of providing a verification service, with set objectives in an attempt to address the problem stated. Hence, it adopted the SoaML in order to analyze and design artifact for the proposed system.

A web services enabled system for verifying document was built alongside with a web application for consuming the service provided by the web services and an interface for uploading document details for a selected few institutions.

The service receives a request from the client side and sends the request to the appropriate department based on the parameter that was passed, the request is then processed and the result is sent to the output screen on the client side.

An interface was built for all the institutions, highlighted as the case study, admins to upload document details of fresh graduates to the appropriate database thereby achieving the aim and objectives of this project.

The service was built with Python using UNIBEN, AAU and UNILAG as a case study. It employed the use of distributed databases where each institution could be viewed as an institution with a digital repository to reduce the risk associated with a centralized database like that of LASU.

5.2 Conclusion

Document fraud has eaten deep into our academic institutions and the increase knows no bound due to the poor systems for verification. Upon this, this research work developed a prototype system for combating academic certificate fraud in tertiary institutions. The system however, integrates distributed document repository with the interoperable capabilities of web services. There is now a reduced difficulty in document verification for both the institution and prospective recruiting firm.

5.3 Recommendation

This study attempts to reveal the impact of Document Verification Service in the Nigerian Education System. The Document Verification Service is suggested to the Nigerian University Commission for adoption and integration as it would assist in detecting forged documents, reduce the time, cost, risks and other problems involved in verifying documents and enhance the global standard of Nigerian degree documents.

REFERENCES

Adjei J. et al. (2014), “Document Authentication System Preventing and Detecting Fraud of Paper Documents”

Afrakhteh M., Ibrahim S., and Salleh M. (2015), “Printed document authentication using watermarking technique”

BellaNaija.com. (2015), bellanaija.com. Retrieved from bellanaija.com https://www.bellanaija.com/2015/10/lagos-banker-arrested-for-using-forged-wasce-university-results-to-gain-employment/

Bernstein, Daniel J. (2020). "Protecting communications against forgery". Retrieved from https://cr.yp.to/antiforgery/forgery-20080501.pdf

Eason R. (2018) “A Model of Unforgeable Digital Certificate Document System”

CIMA (2008). Fraud Risk Management: A Guide to Good Practice, UK. CIMA.

Chen Z., (2018) “Anti-Counterfeit Authentication System of Printed Information Based on a Logic Signing Technique”

Chen-Wilson L. et al., (2018), “Secure certification for ePortfolios

Don, O. (2016, July 26). https://donokereke.blogspot.com.ng. Retrieved from https://donokereke.blogspot.com.ng: https://donokereke.blogspot.com.ng/2016/07/staggering-fake-certificates-fraud-in.html

CIMA. (2018). Fraud risk management: A Guide to Good Practice. UK: CIMA.

Eldefrawy M. H., Alghathbar K., and Khan M. K. (2012), “Hardcopy document authentication based on public key encryption and 2D barcodes”

Elvesaeter, B. (2011), “Service-oriented Heterogenous Architecture and Platforms – Service Modelling with SoaML”, SINTEF ICT, NETsec Software Solutions. 20 pp.

ETX-NG. (2013). About-ext-ng. Retrieved from ETX-NG: www.etx-ng.com/about-etx-ng

Fielding RT (2015). Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California.

Garwe, E. C. (2015). Qualification, Award and Recognition Fraud in Higher Education in Zimbabwe. Macrothink Institute.

Glasner, M, (2015). Keeping in check: Why Background Screening is needed. HR Strategy.

Garain U. and Halder B. (2008), "On automatic authenticity verification of printed security documents"

Grolleau, G., Lakhal, T., & Mzoughi, N. (2008). An Introduction to the Economics of Fake Degrees. Journal of Economic Issues.

Havocscope. (2014). Global Black Market Information. Retrieved May 31, 2021, from https://www.havocscope.com/tag/kenya/

HEDD. (2016). Advice and Guidance on Degree Fraud. A Toolkit for Employers. Manchester: Graduate Prospects, Prospects House.

Kathleen Richards (2020). "What is Cryptography?" Retrieved from: Search Security at https://searchsecurity.techtarget.com/definition/cryptography

Madasu V., Hafizuddin M., Yusof M., and Hanmandlu M. (2003), "Automatic Extraction of Signatures from Bank Cheques and other Documents"

Mavroeidis, Vasileios, and Kamer V. (2018), "The Impact of Quantum Computing on Present Cryptography"

Mojeed, M. (2014). Aviation Minister, Stella Oduah in fresh fake doctorate degree scandal. Retrieved from Premium Times: https://www.premiumtimes.com/news/152958-aviation-minister-stella-oduah-in-fresh-fake-doctorate-degree-scandal.html

Montazerolghaem, Ahmadreza; Yaghmaee, Mohammad Hossein; Leon-Garcia, Alberto (2020). "Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation".

NewsPunch. (2017). Certificate Fraud: Dino Melaye's Name Missing from ABU Alumni Website. Retrieved from NewsPunch: https://www.newspunch.org/2017/03/certificate-fraud-dino-melayes-name.html

Nizar, K. (2013). The penalty for forgery. Retrieved from Gulf Times: https://www.gulf-times.com/story.348045/The-penalty-for-forgery

Ochieng, H. O. (20160. A Mobile based Application for Verification of Legitimacy of Degree Certificate in Kenya (Thesis). Strathmore University SU+ @ Strathmore University Library, 110171/4904.

Oestreich Ken (2015). "Converged Infrastructure". Retrieved from Thecto Forum at: http://www.thectoforum.com/content/converged-infrastructure-0

PremiumTimes, (2016). Buhari’s “no WASC certificate” suit withdrawn. Retrieved May 31, 2021, from Premium Times: https://www.premiumtimesng.com/news/headlines/206130-breakinf-buharis-no-wasc-certificate-suit-withdrawn.html

Ravishankar S., Prashanth Iye R., Balsubramanian S. (2009), “Mark sheet verification Anti-counterfeiting”.

Saharareporters, N. Y. (2017). WAEC Confirms Senator Andy Uba Forged Secondary School Certificate. Retrieved from saharareporters: https://saharareporters.com/2017/03/31/waec-confirms-senator-andy-uba-forged-secondary-school-certificate

Salleh M. and Yew T. (2015), “Application of 2D barcode in hardcopy document verification system”

Sheng and X. Wu (2012), “A new digital anti-counterfeiting scheme based on chaotic cryptography”

Steve Rangers, (2018) “What is cloud computing? Everything you need to know about the cloud explained” Retrieved from: https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/

Ted S., Jason N. (2017), “Hands on Virtual Computing”.

Tunggal, Abi (2020). "What Is a Man-in-the-Middle Attack and How Can It Be Prevented - What is the difference between a man-in-the-middle attack and sniffing?" Retrieved from: https://www.upguard.com/blog/man-in-the-middle-attack#mitm-sniffing

Victor R. (2014) Internet Recruiting Power: Opportunities and Effectiveness. Research paper, International Research Centre on Organizations (IRCO).

Voloshynovskiy S. et al. (2015), "Information-Theoretic Analysis of Electronic and Printed Document Authentication"

APPENDIX

app.py

```
import requests

from flask import Flask, request, render_template, redirect, url_for,
send_from_directory, send_file, make_response

app = Flask(__name__)

uri = ' http://202.61.242.18:8080'

@app.route('/')
def index():
    return render_template('index.html')


@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        url = f'{uri}/admin/signup'
        data = dict(request.form)
        req_signup = requests.post(url, json=data)
        if req_signup.ok:
            return redirect(url_for('login'))
        return req_signup.json()

    return render_template('signup.html')


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        url = f'{uri}/login'
        data = dict(request.form)
        response = requests.post(url, json=data)
        if response.ok:
            r_json = response.json()
            logged_in_as = r_json['logged_in_as']
            return redirect(url_for('verify', logged_in_as=logged_in_as))
        print('unsuccessful')

    return render_template('login.html')


@app.route('/sign', methods=['GET', 'POST'])
def sign():
    if request.method == 'POST':
        url = f'{uri}/sign'
        data_files = request.files
        data_json = request.form
        print(request.files)
        print(request.form)
        response = requests.post(url, files=data_files, data=data_json)
        if response.ok:
            response = make_response(response.content)
            response.headers['Content-Type'] = 'application/pdf'
            response.headers['Content-Disposition'] = \
                'inline; filename=%s.pdf' % 'yourfilename'
            return response
        else:
            return render_template('signdoc.html', data=response.json())

    return render_template('signdoc.html')


@app.route('/verify/<logged_in_as>', methods=['GET', 'POST'])
def verify(logged_in_as='user'):
    url = f'{uri}/verify'
    if 'doc_id' in request.values:
        response = requests.post(url, data=request.values)
        data = response.json()
        if response.ok:
            return render_template('profile.html', data=data)
        elif response.status_code == 404:
            return render_template('invalid.html')
        else:
            return render_template('invalid.html')

    if request.method == 'POST':
        response = requests.post(url, data=request.form)
        data = response.json()
        if response.ok:
            return render_template('profile.html', data=data)
        elif response.status_code == 404:
            return render_template('invalid.html')
        else:
            return render_template('invalid.html')

    return render_template('verifyDoc.html', logged_in_as=logged_in_as)


@app.route('/request', methods=['GET', 'POST'])
def requesttrans():
    if request.method == 'POST':
        school = request.form['institution']

        filename = f'{school}-david-transcript.pdf'
        return send_from_directory('static', path=filename,
                                   as_attachment=True, filename=filename)
    return render_template('requesttrans.html')


@app.route('/gen/transcript', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        return send_from_directory('static', path='uniben-david-transcript.pdf',
                                   as_attachment=True, filename='uniben-david-
transcript.pdf')
```

```
    return render_template('gentranscript.html')


if __name__ == '__main__':
    app.run(debug=True, port=8080)
```

Signup.html

```
<!DOCTYPE html>
<html>
<head>
        <title>DocVerify Signup</title>
        <link rel="stylesheet" href="{{url_for('static', filename='css/reg.css')}}">
    <link href='https://fonts.googleapis.com/css?family=Nunito:400,300'
rel='stylesheet' type='text/css'>
        <link
href="https://fonts.googleapis.com/css?family=Poppins:600&display=swap"
rel="stylesheet">
        <script src="https://kit.fontawesome.com/a81368914c.js"></script>
        <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
    <header>
        <div class="container">
            <nav class="nav d-flex">
                <a href="/" class="logo">
                    <img src="{{url_for('static', filename='images/logo.png')}}"
alt="logo">
                </a>
                <!-- toggle bar -->
                <div class="burger">
                    <span></span>
                    <span></span>
                    <span></span>
                </div>
                <div class="navigation-bar">
                    <ul>
                        <!-- <li><a href="#">Verify Document</a></li>
                        <li><a href="#">Request Transcript</a></li> -->
                        <li><a href="/login" class="btn1">Login</a></li>
                     </ul>
                </div>
            </nav>
        </div>
    </header>
        <!-- <img class="wave" src="img/wave.png">  -->
         <div class="container1">
        <div class="img1">
                          <img src="{{url_for('static',
filename='images/welocm.svg')}}">
                 </div>
        <div class="reg_content">
            <form action="/signup" method="post">
```

```
                <h1>Create An Account</h1>
<!--                <h3>Please Fill in this form to create an account</h3>-->

                <fieldset>
                <label for="name">Name:</label>
                <input type="text" id="name" name="name">

                <label for="mail">Email:</label>
                <input type="email" id="mail" name="email">

                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>


                    <label for="job">Select Institutions:</label>
                    <select id="job" name="institution" required>
                        <option>Ambrose Alli University</option>
                        <option>University of Benin</option>
                        <option>University of Ibadan</option>
                        <option>University of Ilorin</option>
                        <option>University of Lagos</option>
                        <option>University of Port Harcourt</option>
                        <option></option>
                    </select>
                    <label for="job">Select Institutions Code:</label>
                    <select id="job" name="inst_code" required>
                        <option value="">Institution Code</option>
                        <option>AAU</option>
                        <option>UNIBEN</option>
                        <option>UNIBADAN</option>
                        <option>UNILORIN</option>
                        <option>UNILAG</option>
                        <option>UNIPORT</option>
                    </select>
                </fieldset>
                <button type="submit">Sign Up</button>
            </form>
        </div>


    </div>
    <script type="text/javascript" src="js/app.js"></script>
        <script src="{{url_for('static', filename='js/jquery.min.js')}}"></script>
        <script src="{{url_for('static', filename='js/main.js')}}"></script>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html>
<head>
        <title>DocVerify Login</title>
        <link rel="stylesheet" href="{{url_for('static', 
filename='css/style.css')}}">
        <link 
href="https://fonts.googleapis.com/css?family=Poppins:600&display=swap" 
rel="stylesheet">
        <script src="https://kit.fontawesome.com/a81368914c.js"></script>
        <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
    <header>
        <div class="container">
            <nav class="nav d-flex">
                <a href="/" class="logo">
                    <img src="{{url_for('static', filename='images/logo.png')}}" 
alt="logo">
                </a>
                <!-- toggle bar -->
                <div class="burger">
                    <span></span>
                    <span></span>
                    <span></span>
                </div>
                <div class="navigation-bar">
                    <ul>
                        <li><a href="/verify">Verify Document</a></li>
                        <li><a href="#">Request Transcript</a></li>
                        <!-- <li><a href="#" class="btn1">Login</a></li> -->
                    </ul>
                </div>
            </nav>
        </div>
    </header>
        <!-- <img class="wave" src="img/wave.png">  -->
         <div class="container1">
                <div class="img1">
                        <img src="{{url_for('static', filename='images/win 
log.svg')}}">
                </div>
                <div class="login-content">
                        <form action="/login" method="post">
                                <img src="{{url_for('static', 
filename='images/profile.svg')}}">
                                <h2 class="title">Welcome</h2>
                        <div class="input-div one">
                           <div class="i">
                                        <i class="fas fa-user"></i>
                           </div>
                           <div class="div">
                                        <h5>Email</h5>
                                        <input type="text" class="input" 
name="email">
                           </div>
                        </div>
                        <div class="input-div pass">
                           <div class="i">
```

```
                                  <i class="fas fa-lock"></i>
                            </div>
                            <div class="div">
                                  <h5>Password</h5>
                                  <input type="password" class="input" 
name="password">
                    </div>
                 </div>
                 <a href="#" class="a">Forgot Password?</a>
                 <input type="submit" class="btn" value="Login">
            </form>
        </div>
    </div>
    <script type="text/javascript" src="js/app.js"></script>
        <script src="{{url_for('static', filename='js/jquery.min.js')}}"></script>
        <script src="{{url_for('static', filename='js/main.js')}}"></script>
</body>
</html>
```

Sign.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DocVerify</title>
    <link rel="stylesheet" href="{{url_for('static', filename='css/sign.css')}}">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.12.1/css/all.min.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js" 
charset="utf-8"></script>
  </head>
  <body>

    <input type="checkbox" id="check">

    <!--header area start-->
    <header>
      <!-- <label for="check">
        <i class="fas fa-bars" id="sidebar_btn"></i>
      </label> -->
      <div class="left_area">
        <a href="/" class="logo">
                    <img src="{{url_for('static', filename='images/logo.png')}}" 
alt="logo">
                </a>
      </div>
      <div class="right_area">
        <a href="#" class="logout_btn">Logout</a>
      </div>
    </header>
    <!--header area end-->
    <!--mobile navigation bar start-->
    <div class="mobile_nav">
      <div class="nav_bar">
        <i class="fa fa-bars nav_btn"></i>
      </div>
      <div class="mobile_nav_items">
        <a href="#"><i class="fas fa-desktop"></i><span>Sign </span></a>
        <a href="/verify/admin"><i class="fas fa-cogs"></i><span>Verify</span></a>
        <a href="/gen/transcript"><i class="fas fa-table"></i><span>Generate 
Transcipt</span></a>
      </div>
    </div>
    <!--mobile navigation bar end-->
    <!--sidebar start-->
    <div class="sidebar">

      <a href="#"><i class="fas fa-desktop"></i><span>Sign</span></a>
      <a href="/verify/admin"><i class="fas fa-cogs"></i><span>Verify</span></a>
      <a href="/gen/transcript"><i class="fas fa-table"></i><span>Generate 
Transcript</span></a>

    </div>
    <!--sidebar end-->

    <div class="container1">

      <div class="reg-content">
        <form action="/sign" method="post" enctype="multipart/form-data">
          <div class="content">
              <h2 >Sign Document</h2>
              {% if data %}
                <p style="color:red;">Document with ID {{data.doc_id}} exists. Verify 
Doc
                    <a style="color:red;" href="{{ request.host_url 
}}/verify/admin?doc_id={{ data.doc_id }}">here</a></p>
              {% endif %}
        <div class="column">
              <div class="card">
                <label for="key">Full Name:</label>
                    <input  type="text" name="fullname" id="name" 
placeholder="Fullname"/>
                <label for="key1">Course:</label>
                    <input  type="text" name="course" id="course" 
placeholder="Course"/>
                <label for="key1">Department:</label>
                    <input  type="text" name="department" id="dept" 
placeholder="Department"/>
                <label for="key">Faculty:</label>
                    <input  type="text" name="faculty" id="fac" 
placeholder="Faculty"/>
                </div>

                <div class="card">
                    <label for="key">Graduation Year:</label>
                    <input  type="text" name="graduation_year" 
id="key"placeholder="Graduation Year"/>
                <label for="key">Matric No:</label>
                    <input  type="text" name="matric_no" id="mat"placeholder="Matric 
No"/>
                <label for="key">Class of Degree:</label>
                    <input  type="text" name="class_of_degree" id="key" 
placeholder="Class of Degree"/>
                <label for="key">Document ID Number:</label>
                    <input  type="text" name="doc_id" id="key" placeholder="Document 
Id"/>
            </div>
            </div>
              <div class="card1">
                  <label for="key">Upload Photograph:</label>
                      <input class="upload" type="file" name="photograph"/>
              </div>
              <div class="card1">
                  <label for="key">Upload Document:</label>
                      <input class="upload" type="file" name="document"/>
              </div>
              <div>
                  <button class="form_btn" >Sign Document</button>
              </div>
          </div>
        </form>
      </div>
      <div class="img1">
        <img src="{{url_for('static', filename='images/addfile.svg')}}">
      </div>
    </div>



    <script type="text/javascript">
    $(document).ready(function(){
      $('.nav_btn').click(function(){
        $('.mobile_nav_items').toggleClass('active');
      });
    });
    </script>

  </body>
</html
```

Verify.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Verify Document</title>
    <link rel="stylesheet" href="{{url_for('static', filename='css/sign.css')}}">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.12.1/css/all.min.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"
charset="utf-8"></script>
  </head>
  <body>

    <input type="checkbox" id="check">

    <!--header area start-->
    <header>
      <!-- <label for="check">
        <i class="fas fa-bars" id="sidebar_btn"></i>
      </label> -->
      <div class="left_area">
        <a href="/" class="logo">
            <img src="{{url_for('static', filename='images/logo.png')}}" alt="logo">
        </a>
      </div>
      <div class="right_area">
        <a href="#" class="logout_btn">Logout</a>
      </div>
    </header>
    <!--header area end-->

    <!--mobile navigation bar start-->
    <div class="mobile_nav_items">
        {% if 'admin' == logged_in_as %}
        <a href="/sign"><i class="fas fa-desktop"></i><span>Sign</span></a>
        <a href="#"><i class="fas fa-cogs"></i><span>Verify</span></a>
        <a href="/gen/transcript"><i class="fas fa-table"></i><span>Generate
Transcipt</span></a>
        {% endif %}
        {% if 'user' == logged_in_as %}
        <a href="/request"><i class="fas fa-desktop"></i><span>Request Transcript
</span></a>
        <a href="#"><i class="fas fa-cogs"></i><span>Verify</span></a>
<!--        <a href="/upload"><i class="fas fa-table"></i><span>Upload
Transcipt</span></a>-->
        {% endif %}
      </div>
    </div>
    <!--mobile navigation bar end-->
    <!--sidebar start-->
    <div class="sidebar">

      {% if 'admin' == logged_in_as %}
        <a href="/sign"><i class="fas fa-desktop"></i><span>Sign</span></a>
        <a href="#"><i class="fas fa-cogs"></i><span>Verify</span></a>
        <a href="/gen/transcript"><i class="fas fa-table"></i><span>Generate
Transcipt</span></a>
        {% endif %}
        {% if 'user' == logged_in_as %}
        <a href="/request"><i class="fas fa-desktop"></i><span>Request
Transcript</span></a>
```

```
        <a href="#"><i class="fas fa-cogs"></i><span>Verify</span></a>
<!--        <a href="/upload"><i class="fas fa-table"></i><span>Upload
Transcipt</span></a>-->
        {% endif %}

    </div>
    <!--sidebar end-->
    <div class="container1">

          <div class="reg-content">
            <form action="/verify/{{ logged_in_as }}" method="post"
enctype="multipart/form-data">
              <div class="content">
                <h2>Verify Document</h2>
                  <div class="card">
                      <label for="key">Please Enter Document ID:</label>
                          <input  type="text" name="doc_id" placeholder="Document
ID"/>
                  </div>

                  <div>
                      <button class="form_btn" >Verify Document</button>
                  </div>
              </div>
          </form>
        </div>
        <div class="img1">
          <img src="{{url_for('static', filename='images/pull re.svg')}}">
        </div>
     </div>


    <script type="text/javascript">
    $(document).ready(function(){
      $('.nav_btn').click(function(){
        $('.mobile_nav_items').toggleClass('active');
      });
    });
    </script>

  </body>
</html>
```

Profile.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Student Valid</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
        <script src="https://code.jquery.com/jquery-1.10.2.min.js"></script>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/js/bootstrap.bundle.min.js"></
script>
</head>
<body>
<div class="container">
    <div class="main-body">
      <a href="/" class="logo">
        <img src="{{url_for('static', filename='images/logo.png')}}" alt="logo">
      </a>
      <p>The following are information of the student of which document ID was
provided for Verification.

      This below details is proof that the student document is Valid.<p>
          <div class="row gutters-sm">
            <div class="col-md-4 mb-3">
              <div class="card">
                <div class="card-body">
                  <div class="d-flex flex-column align-items-center text-center">
                    <img src="http://localhost:5000/student/{{ data['doc_id']
}}/image" class="rounded-circle" width="150">
                    <div class="mt-3">
                      <h4>{{ data['fullname'] }}</h4>
                    </div>
                  </div>
                </div>
              </div>
            </div>
            <div class="col-md-8">
              <div class="card mb-3">
                <div class="card-body">
                  <div class="row">
                    <div class="col-sm-3">
                      <h6 class="mb-0">Full Name</h6>
                    </div>
                    <div class="col-sm-9 text-secondary">
                      {{ data['fullname'] }}
                    </div>
                  </div>
                  <hr>
                  <div class="row">
                    <div class="col-sm-3">
                      <h6 class="mb-0">Matric Number</h6>
                    </div>
                    <div class="col-sm-9 text-secondary">
                      {{ data['matric_no'] }}
                    </div>
                  </div>
                  <hr>
                  <div class="row">
                    <div class="col-sm-3">
                      <h6 class="mb-0">Course</h6>
                    </div>
                    <div class="col-sm-9 text-secondary">
                      {{ data['course'] }}
          </div>
         </div>
         <hr>
         <div class="row">
          <div class="col-sm-3">
           <h6 class="mb-0">Department</h6>
          </div>
          <div class="col-sm-9 text-secondary">
           {{ data['department'] }}
          </div>
         </div>
         <hr>
         <div class="row">
          <div class="col-sm-3">
           <h6 class="mb-0">Faculty</h6>
          </div>
          <div class="col-sm-9 text-secondary">
           {{ data['faculty'] }}
          </div>
         </div>
         <hr>
         <div class="row">
          <div class="col-sm-3">
           <h6 class="mb-0">Class of Degree</h6>
          </div>
          <div class="col-sm-9 text-secondary">
           {{ data['class_of_degree'] }}
          </div>
         </div>
         <hr>
         <div class="row">
          <div class="col-sm-3">
           <h6 class="mb-0">Graduation Year</h6>
          </div>
          <div class="col-sm-9 text-secondary">
           {{ data['graduation_year'] }}
          </div>
         </div>
         <hr>
        </div>
       </div>

      </div>
     </div>

    </div>
  </div>

<script type="text/javascript">

</script>
</body>
</html>
```

RequestTranscript.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Request a Transcript</title>
    <link rel="stylesheet" href="{{url_for('static', filename='css/reqTrans.css')}}">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.12.1/css/all.min.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"
charset="utf-8"></script>
  </head>
  <body>

    <input type="checkbox" id="check">

    <!--header area start-->
    <header>
      <!-- <label for="check">
        <i class="fas fa-bars" id="sidebar_btn"></i>
      </label> -->
      <div class="left_area">
        <a href="/" class="logo">
                    <img src="{{url_for('static', filename='images/logo.png')}}"
alt="logo">
                </a>
      </div>
      <div class="right_area">
        <a href="#" class="logout_btn">Logout</a>
      </div>
    </header>
    <!--header area end-->

    <!--mobile navigation bar start-->
    <div class="mobile_nav">
      <div class="nav_bar">
        <i class="fa fa-bars nav_btn"></i>
      </div>
      <div class="mobile_nav_items">
        <a href="#"><i class="fas fa-desktop"></i><span>Request Transcript
</span></a>
        <a href="/verify/user"><i class="fas fa-cogs"></i><span>Verify</span></a>
      </div>
    </div>
    <!--mobile navigation bar end-->
    <!--sidebar start-->
    <div class="sidebar">

      <a href="#"><i class="fas fa-desktop"></i><span>Request Transcript</span></a>
      <a href="/verify/user"><i class="fas fa-cogs"></i><span>Verify</span></a>
    </div>
    <!--sidebar end-->

    <div class="container1">
        <form action="/request" method="post">
            <div class="content">
                <h2>Request For Transcript</h2>
                    <div class="card">
                        <label for="name">Matric Number:</label>
                        <input type="text" id="name" name="matno">
                    </div>
                    <div class="card">
                        <label for="name">Year of Graduation:</label>
                        <input type="text" id="name" name="year">
                    </div>
                    <div class="card">
                        <label for="job">Select Institutions:</label>
                        <select id="job" name="institution" required>
                            <option value="">Select Institution</option>
                            <option value="aau">Ambrose Alli University</option>
                            <option value="uniben">University of Benin</option>
                            <option value="unilag">University of Lagos</option>
                            <option></option>
                        </select>
                    </div>
                <button type="submit">Submit</button>
        </form>
        </div>

      <div class="img1">
          <img src="{{url_for('static', filename='images/auth.svg')}}">
      </div>
    </div>

    <script type="text/javascript">
    $(document).ready(function(){
      $('.nav_btn').click(function(){
        $('.mobile_nav_items').toggleClass('active');
      });
    });
    </script>

  </body>
</html>
```