INTELLIGENT AGENT SERVICE FOR DETECTING IMPERSONATORS IN ONLINE EXAMINATION ENVIRONMENT USING K-NEAREST NEIGHBOR ALGORITHM

BY

IJEOMA ONYINYE IZU-OKPARA (B.Sc.)

(20095775799)

A THESIS PRESENTED TO POSTGRADUATE SCHOOL,
FEDERAL UNIVERSITY OF TECHNOLOGY, OWERRI

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF TECHNOLOGY (MTECH) IN INFORMATION MANAGEMENT TECHNOLOGY

OCTOBER, 2020

CERTIFICATION

I certify that this research "**Intelligent Agent Service for Detecting Impersonators in Online Examination Environment using K-Nearest Neighbor Algorithm**" was carried out by **Ijeoma Onyinye Izu-Okpara** (20095775799) in partial fulfillment of the requirements for the award of the Master of Technology (MTech) in Information Management Technology of Federal University of Technology Owerri.

Engr. Dr. O. C. Nwokonkwo
(Supervisor)

16/11/2022
Date

Dr. A. I. Otuonye
(Ag. Head of Department)

18/11/22
Date

Prof. (Mrs) U. F. Eze
(Dean of SICT)

30/11/22
Date

Prof. B. O. Esonu
(Dean, Postgraduate School)

Date

Prof. V. E. Ejiofor
(External Examiner)

06/12/2022
Date

DEDICATION

This Work is dedicated first to God Almighty for His strength and Grace and to my wonderful spouse for his amazing support.

ACKNOWLEDGEMENT

There are some exceptional people without whom this work would not have seen the light of day. Special thanks to Engr. Dr. O. C. Nwokonkwo, our Head of Department, who gave me the privilege of having him as a supervisor; Dr. Charles Ikerionwu who gave me the initial push, persistent motivation and technical backup to set the ball rolling; and Dr. John-Otumu Adetokunbo who was relentless in his professional guidance and assistance whenever I needed it, and other lecturers such as Prof. Mrs. U. F. Eze, Engr. Dr. E. C. Amadi, Dr. A. I. Otuonye, Dr. N. Ekedebe, Dr. P. A. Alabi and others too numerous to mention. May God reward you all.

ABSTRACT

Intelligent Agent Service for Detecting Impersonators in Online Examination Environment was developed for managing major challenges such as security and cheating (impersonation) that is now a critical issue associated with online examination system or computer-based testing (CBT). The key informant interview technique, observations, and critical review of articles related to CBT methodologies (design and development) were used to gather facts regarding the study area. The agile software methodology was adopted as the software development life cycle based on its strengths in team work and efficient product delivery haven examined six different software methodologies. A multi-level security service was developed to handle various security threats at the different operational levels of the proposed system using 256-bit SSH algorithm, Merssene Twister Algorithm, 128-bit Advanced Encryption Standard (AES), and Message Digest (MD) 5 algorithm. The K-Nearest Neighbor (KNN) machine learning algorithm was implemented as an intelligent agent service to give the proposed system some intelligence in detecting and classifying a likely suspected case of impersonation and its severity level during an online examination. JavaScript, Hypertext preprocessor (PHP), MySQL, Hypertext Markup Language (HTML), Cascading Style Sheet (CSS) and Python programming language were used to develop the software prototype. Unified Modeling Language (UML) such as sequence diagram and use-case diagram were used to model the system behavior and interactivity. The developed system accuracy in terms of the algorithm used for detecting impersonators was evaluated using confusion matrix. Results revealed 98% accuracy with the K-NN algorithm implemented. Finally, the results based on the acceptance testing done also revealed that 86% of the users strongly agreed with the performance level of the developed platform; hence, recommendations is made for the proposed system to be adopted by Nigerian Universities and companies based on its effectiveness in impersonation detection.

Keywords: K-NN Algorithm, Classification, Intelligence, Online Examination, Agile Model

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

1.1 Background Information

Teaching and learning are repeatedly being moved to different technological platforms (Abubakar and Adebayo, 2014). The World Wide Web (www) has become a very powerful and useful tool for the administration of teaching and learning. It is noted that this giant stride in technological advancement has necessitated the move of educational resources and assessment of students by educational institutions and certification companies by the adoption of computer-based testing (CBT) systems for entrance into Nigeria institutions of higher learning (Sadiq and Onianwa, 2011).

According to Jimoh, AbdulJaleel and Kawu (2012), technology has drastically changed the pattern of assessing students over the past years. Assessment being a very important part of educational process that cannot be quickly put aside (Delen, 2015). The advancement of information technology (IT) over the last few decades has made education more interesting by conducting examination for students and assessing them via CBT (Abedi, 2014; Chua, 2012, Chou, Moslehpour and Huyen, 2014).

Examination is referred to a common use and acceptable factor for assessing or evaluating student's understanding and cognitive aptitude of a topic or subject matter he or she was taught (Ajinaja, 2017). Examination is also referred to as the greatest technique of appraising the understanding and skills of an individual under a specific condition (Adebayo and Abdulhamid, 2009; Ismail and Soye, 2018).

Ismail and Shami (2012) is of the opinion that examinations can also regulates the degree to which instructive purposes have remained accomplished as well as the magnitude to which enlightening establishments have helped the requirements of public and humanity at-large. According to Adebayo and Abdulhamid (2009), different means have been utilized over the years to conduct and examine the abilities of student during under examination conditions; these ranges from manual pen and paper-based, verbal to in-print, applied to hypothetical, etc.

Fagbola, Adigun, and Oke (2013) opined that the manual pen and paper-based examination has existed for a very long time and it is fast becoming an obsolete pattern of writing examination in modern times due to the numerous challenges associated with it such as cost implication for production of writing materials, examination venue allocation issues, examination malpractices like impersonation, time taken to release results after examination, poor comfort of students writing examination, human mistakes, and so on.

According to Kuyoro, Maminor, Kanu, and Akande (2016), the practice of examinations for evaluation started over a thousand (1,000) years ago; it was said to have been used for the engagement and elevation of employees in the Chinese public service which took place during the Han Empire in 207 BC. The test encompassed the philosophy of Confucius and subjects such as calligraphy, geography, mathematics, and poetry.

Presently, the rapid advancement of Information and Communication Technologies (ICT) has created a new paradigm-shift for teaching and learning over computer network environments which has indeed necessitated a serious move from the manual pen and paper type of examination to Computer-Based Testing (CBT), which can also be referred to as E-Assessment, Computer-Based Examination (CBE), E-Examination, Computer

Assisted Testing (CAT), Computer Aided Assessment (CAA), Computer Based Assessment (CBA), based on the challenges associated with the manual pen and paper type of examinations (Ismail and Shami, 2012; Delen, 2015). Computer-Based Testing (CBT) is the usage of information technology (IT) for examination or assessment and associated actions. It permits educationalists and instructors to set and program surveys, questions experiments and examinations to be assembled, administered and appraised through computer systems that are networked together (Omotehinwa and Durojaye, 2013; Ajinaja, 2017; Adebayo and Abdulhamid, 2009; Ismail and Soye, 2018; Fagbola, Adigun, and Oke, 2013; Kuyoro *et al*., 2016).

In recent times, examination bodies such as Joint Admission and Matriculation Board (JAMB), the body in-charge of admitting students into Nigerian universities had adopted the usage of CBT for conducting entrance examination into universities instead of the old pen and paper type (PPT) of examinations (Sadiq and Onianwa, 2011; Ajinaja, 2017).

Similarly, so many Universities in Nigeria had also embarked on the full deployment of CBT centres for conducting their internal examinations and Post-Unified Tertiary Matriculation Examination (Post-UTME) online (Ajinaja, 2017). According to Kuyoro *et al*. (2016), some of the universities that had deployed the usage of CBT for the conduct of examination either partially or fully are listed below:

(i) National Open University of Nigeria (NOUN), Abuja

(ii) University of Ilorin, Kwara State

(iii) Federal University of Technology, Minna, Niger State

(iv) University of Lagos, Akoka, Lagos State

(v) University of Benin, Benin city, Edo State

(vi) Ambrose Alli University, Ekpoma, Edo State

(vii) University of Nigeria, Nsukka, Enugu State

(viii) Covenant University, Ota, Ogun State

(ix) University of Ibadan, Oyo State

(x) Obafemi Awolowo University, Ile-Ife, Osun State

In addition, some secondary schools have also partially or fully executed online examination using CBT (Ajinaja, 2017). Some of the secondary schools includes the following:

(i) Sunshine Secondary School

(ii) Hallmark College, Ibadan, Oyo State

These institutions are happy and waxing stronger in the usage of CBT for examination because the effectiveness of CBT platforms depends greatly on security, standardization, appreciable cost of deployment, and mode of administering the examination. It is noted that CBT also referred to as computer-based examination (CBE), e-assessment, e-examination has become very prevalent and it offers many advantages when deployed and used for conducting examinations (Clariana and Wallace, 2002; Schatz and Putz, 2006; Delen, 2015). Some very useful advantages offered by CBT are as follows: testing can be slated at any time and can be done from anywhere (Jeong, 2014), increased testing capacity of participants (Chua and Don, 2013), provision of very few human resources and reduced paper work (kaya and Delen, 2014; Schatz and Browndyke, 2002).

1.2 Problem Statement

The design, development, deployment and management of CBT in the conduct of online examinations in a global sense have been highly recognized and appreciated (Jimoh, AbdulJaleel and Kawu, 2012). For close to two decades researchers in the field of computing and information technology have provided very powerful platforms for the conduct and assessments of examinations online, which far outweigh the conventional

pen and paper practices done in classrooms (Fagbola, Adigun and Oke, 2013; Ajinaja, 2017; Abdulkareem and Nachndiya, 2018; Adebayo and Abdulhamaid, 2009; Kuyoro et al., 2016; Oluwole, 2015; Ismail and Soye, 2018; Omotehinwa and Durojaye, 2013).

However, the major points of concerns this research work intends to handle are in the areas of monitoring students or examinees against the problems of cheating (impersonation) during the online examination as noted by (Al-Saleem and Uttah, 2014) and also the problem of security at different levels while using the CBT platform for conducting online examinations (Adebayo and Abdulhamaid, 2009; Ayo, Akinyemi, Adebiyi and Ekong, 2007).

It is against these backdrops that this research work is aimed at achieving the three major objectives as itemized in section 1.3, in order to effectively tackle the problems identified.

1.3 Objectives

The broad objective of this study is to design a smart and secure computer based testing platform for the effective conduct of online examinations and monitoring of students against impersonation without human supervision.

The specific objectives of the study are to:

(a) develop a prototype CBT platform for conduct of online examination using the standard Triple-A model (assembling, administering, appraising)

(b) develop a multi-layer based cryptographic services for securing the CBT platform using 256-bit SSH algorithm, Merssene Twister Algorithm, 128-bit Advanced Encryption Standard (AES) alongside with the IPSec, and the Message Digest 5 algorithm (MD5).

(c) develop an intelligent agent service for detecting impersonation threat and classification of the threat into its severity using K-NN machine learning classification algorithm.

1.4 Research Questions

The following research questions were well articulated in addressing the problem statement identified in this study.

(a) Can the CBT platform conduct online examination effectively?

(b) Can the CBT platform secure user's access, and data integrity effectively?

(c) Can the CBT platform be smart enough to effectively detect and prevent a suspected case of impersonator during an online examination?

1.5 Justification of the Study

The shortfall from the research works conducted by (Fagbola, Adigun and Oke, 2013; Ajinaja, 2017; Abdulkareem and Nachndiya, 2018; Adebayo and Abdulhamaid, 2009; Kuyoro et al., 2016; Oluwole, 2015; Ismail and Soye, 2018; Omotehinwa and Durojaye, 2013) on online examination systems or computer based testing systems really motivated and justified this research work in the areas of security with respect to impersonation in online examination environment.

1.6 Scope of the Study

The scope of this study is limited to the conduct of online examination over the intranet only. The proposed intelligent agent service for detecting impersonators in online examination was designed and developed to conduct linear-type of testing. The proposed system can also perform the following: flexible delivery of examination administration through its powerful user's interface/dashboard, data-rich examination results, multi-layer based enhanced security, enhanced presentation features, availability of instant scoring on examination completion or expiration of examination time, enhanced examination results

and data collection window, and detection of possible cases of impersonators during the online examination.

However, the proposed system cannot be used for conducting examination over the Internet; this is based on the curbing out-bound security threat out rightly. It can also not perform the following: adaptive-type of examination, audio/video-based examination questions, and language translation.

CHAPTER TWO
LITERATURE REVIEW

2.1 Conceptual Framework

This section reviewed the general concepts of computer-based testing systems. The section also discusses about the historical and developmental trends in the study area.

2.1.1 Overview of Online Examination System

Online examination system can be described from different researchers' perspective. Zhenming et al. (2013) sees online examination system also known as computer-based testing (CBT) system as a web-based examination system that is suitable for conducting mass assessment in an educational system such as a university. Computer-based testing (CBT) or computer-based assessment is seen as a technology that can bring about change and transformation in learning (Ajinaja, 2017). CBT can also be referred to as a form of examination or testing process in which personal computers are used over a computer network to deliver questions, respond to questions, act as a storage, marking of responses and results reporting (Kuyoro *et al*., 2016). Similarly, Suleiman and Nachandiya (2018), refers to CBT as a form of assessment done using networked computers.

In the last few years, CBT has become one of the most ideal and practical methods of conducting test and examinations in both secondary schools and higher institutions over the conventional pen and paper type of examinations (Suleiman and Nachandiya, 2018). Despite some few notable challenges associated with CBT, many organizations and institutions has adopted the CBT method for evaluation based on its effectiveness in computations and efficient services delivery.

According to Taylor (1998), CBT could be deployed as a desktop-based application over a local area network (LAN) or a web-based application over the Internet or intranet using

web-based technologies. There are basically two types of CBT system for conducting test and examinations as noted by (Ajinaja, 2017; Kuyoro *et al*., 2016; Suleiman and Nachandiya, 2018).

i. Linear-Test

ii. Adaptive-Test

The linear test is a form of test that involves the complete or full length of examination in which the CBT platform randomly selects different questions for different individuals without any preference on their performance level; while in adaptive test, the CBT system deployed actually selects a range of questions from a possible large bank or pool of questions based on individual performance level. It is observed that the selected questions are also classified by contents and difficulty levels.

Researchers such as Ajinaja (2017), Suleiman and Nachandiya (2018), Ismail and Soye (2018), Fagbola, Adigun and Oke, 2013, Oluwole (2015), Adebayo and Abdulhamaid (2009), Rashad et al. (2010) and Ayo et al. (2007) has designed and developed varieties of CBT applications using the linear-test type based on it uniqueness and suitability for institutional peculiarity in examination conduct, having some common notable features listed below:

(i) Upload of questions and corresponding answers

(ii) Students login authentication

(iii) Examination timer

(iv) Randomized questions

(v) Different type of questions (MCQ, objectives, fill in the gaps, etc)

(vi) Auto-marking and grading

(vii) Flexible result checker

(viii) Registration of students

(ix) Update of students information

(x) Event logs

(xi) Reporting modules

(xii) Mass auto-submission

Despite the laudable features in the CBT systems, there are still some visible challenges as listed below:

(i) Multiple security issues (Adebayo and Abdulhamaid, 2009)

(ii) Questions type

(iii) Lack of resumption capability whenever there is power outage or network failure (Younis abd Hussein, 2015)

(iv) Results integrity issues (Adebayo and Abdulhamaid, 2009)

(v) Introduction of errors during upload of questions (Osang, 2009; Valenti, Neri and Cucchiarelli, 2003)

(vi) Cheating (Adebayo and Abdulhamaid, 2009; Kuyoro *et al*., 2016)

However, the CBT systems is far more effective than the conventional pen and paper type in terms of mass conduct of examination, marking, results reporting etc (Kuyoro *et al*., 2016).

2.1.2 Overview of the historical development in CBT systems

Historically, the development of computer-based testing (CBT) system started in the early 1970s with the introduction of personal computers which brought out the usefulness of technology in the areas of teaching, learning and conduct of examination online (Hedberg, Davis, Lundeen and Vitelia, 1995; Greiff and Martin, 2015). According to Kuyoro *et al*. (2016), the Plato system originated the e-assessment from the University of Illinois, and was later turned to commercial product by the Control Data Corporation in

1970s beginning with a computer testing system for National Association of Securities Dealers (NASD), which is now referred to as the Financial Industry Regulatory Authority (FIRA). Similarly, Kuyoro *et al*. (2016) also recorded despite the fact that the computer-based testing or e-assessment business slowly expanded, but it rather paved way for industry-testing bodies like the Thomson Prometric to be recognized as a certified computer-based testing body worldwide.

Further development and growth of the computer-based testing systems that could conduct examination over the Internet also came to the lime light in 1994 by Pearson VUE (Adegbiga, Fakomogbon and Daramola, 2012). The rapid transformation in information and communication technology (ICT) has also birthed different CBT systems by research scholar in a bid to proffer solutions to some of the existing CBT system challenges (Ayo *et al*., 2007; Kuyoro *et al*., 2016; Fagbola, Adigin and Oke, 2013; Ajinaja, 2017; Suleiman and Nachandiya, 2018; Ismail and Soye, 2018; Rashad *et al*., 2010; Alabi *et al*., 2012; Tasci *et al*., 2014).

2.2 Theoretical Framework

This section discusses some major theories, principles, architectural frameworks and models that critically relates to computer-based testing systems.

2.2.1 The Classical Time-Score Test Theory

According to Scheuermann and Pereira (2008), CBT can be seen as an instrument for change, and this change can seriously involve transformation in learning and curricula development in educational institutions. In order to establish a valid computer-based testing (CBT); the International Guidelines on Computer-Based and Internet Delivered Testing (IGCBID) states in 2004 that equivalent test scores should be established for tests

using the pen and paper type mode of examination and the computer-based mode of examination.

According to Allen and Yen (1979), testing standards is supported by the classical true-score test theory, which forms the basis of both the pen and paper type and the computer-based testing type. The classical time-score test theory actually states that there is a likely chance of obtaining same or almost the same test score or result if a given student or examinee takes the same examination in both pen and paper type and in computer-based testing modes. In other words, there might be very little or no significant differences in the test score or result of both type of examination taken.

2.2.2 The Multi-Layered Architecture

According to Cristobal, Sebastian, and Paul (2009), multi-layered architecture is a type of client/server architecture in which the presentation, application or middleware and database tasks are tangibly disjointed. The most prevalent usage of the multi-tier architecture is the three-tier architecture.

Similarly, Ayo et al. (2007) and Akinsanmi, Agbaji, and Soroyewun (2010) revealed that most CBT systems are designed using the 3-tier architecture which comprises the presentation, the logic and the database tiers respectively. The presentation tier is the interface through which the user interacts with the system, the logic tier serves as an engine for processing the users' requests over the network and finally, the database tier which serves as a pool or bank for storing users' information, questions, answers, and results.

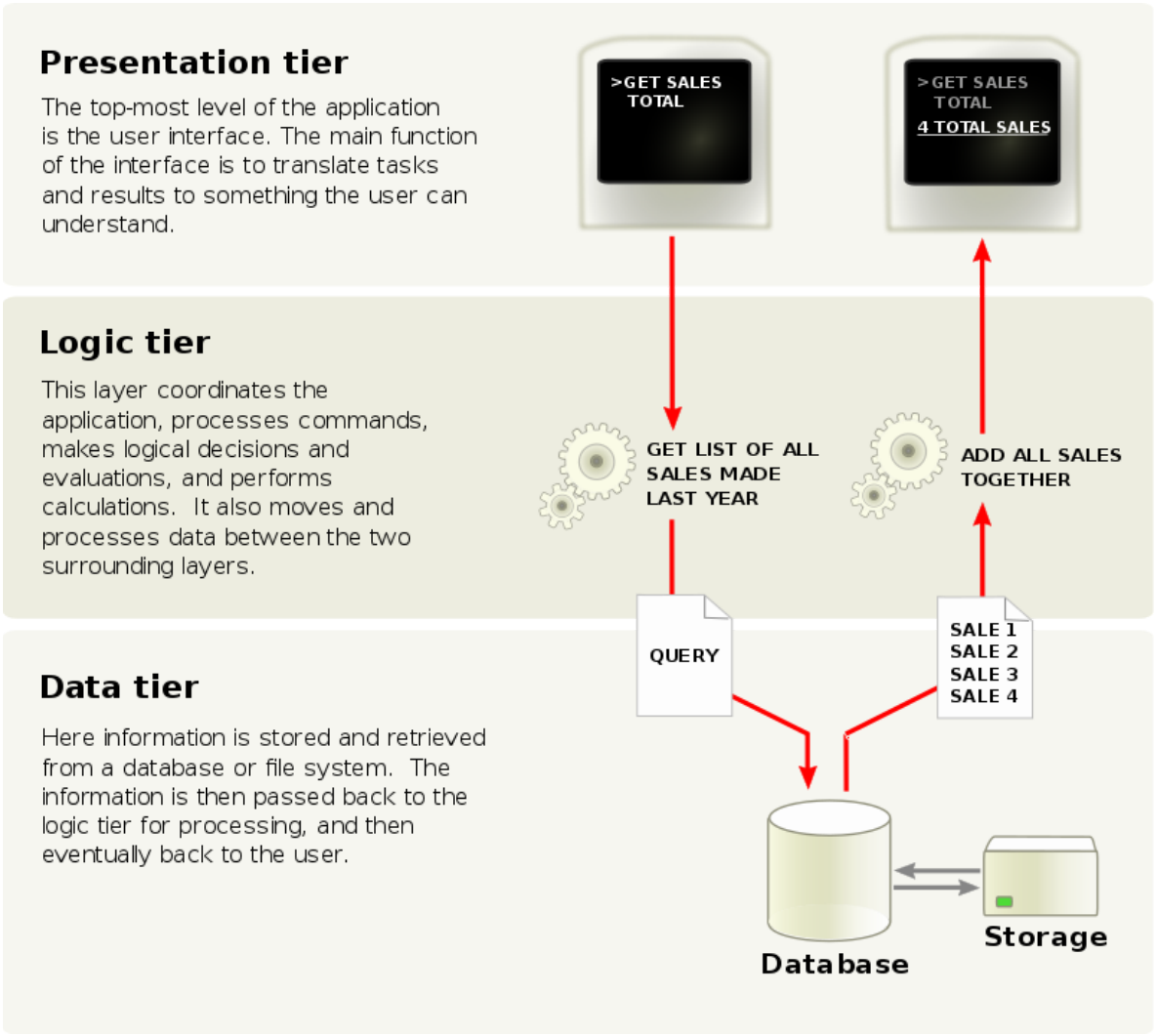

Figure 2.1: Overview of a 3-Tier Architecture (Wikipedia, 2020)

2.2.3 Data Flow Diagram of the CBT System

A data flow diagram uses some primitive symbols to represent the flow of data/information in and out an information system (Ajinaja, 2017). It is observed that only three key players i.e. students, teachers and administrator mainly interacts with most of the existing computer-based testing (CBT) systems in order to conduct examination (Fagbola, Adigun and Oke, 2013; Ajinaja, 2017).

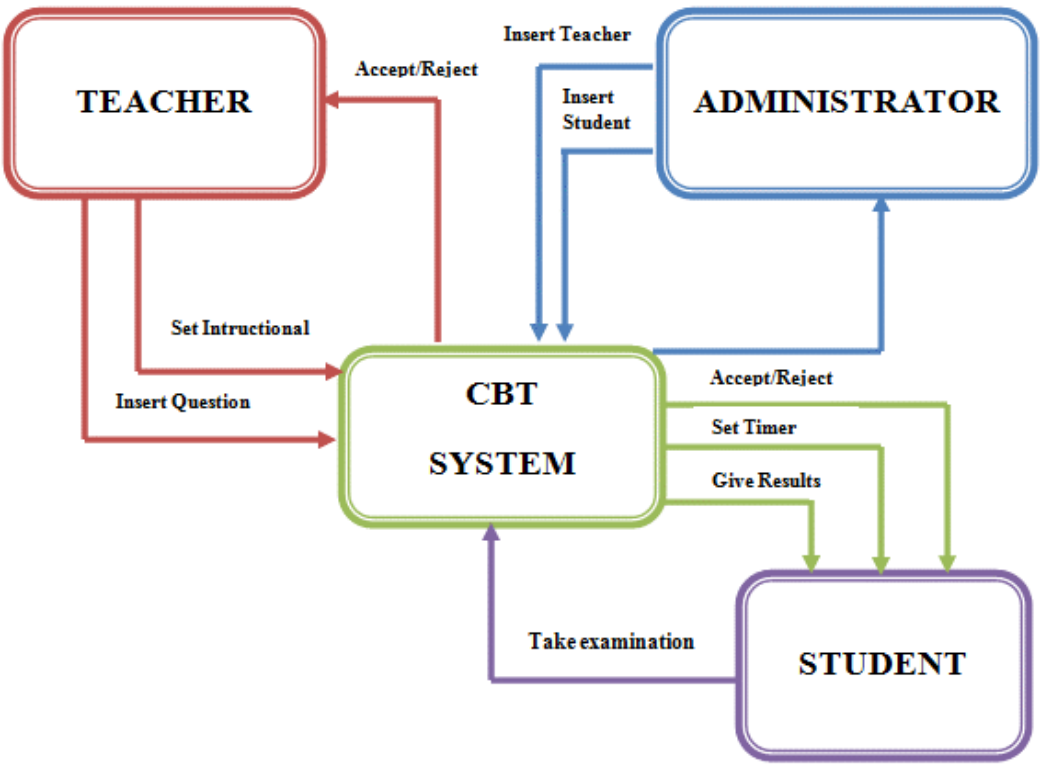

Figure 2.2: Data Flow Diagram of Existing CBT System
(Fagbola, Adigun and Oke, 2013; Ajanaja, 2017)

2.2.4 The Triple-A Model Architecture for Developing CBT

According to Wang et al. (2004) every computer-based testing or e-assessment system should adopt the Triple-A Model which stands for "assembling, administering and appraising" as the reference point qualification in order to deliver the most inclusive form of examination or test, and to be extra appropriate for teacher's education. Figure 2.3 revealed the Triple-A model architecture that have used by different CBT system which consists of three major roles:

(i) **Assembling:** means to build the questions and answers. Firstly, the teachers needs to create or select the course to be evaluated by the test; followed by creating the questions and their answers. Finally, group questions and their answers into a tests or examination format.

(ii) **Administering:** means to deliver or show the examination or test to the students or examinees to enable them execute it. More so, to gather and record the test scores data.

(iii) **Appraising:** means to examine the collected score data of the examination or test and to further produce report.

The teachers can then select a specific test or examination item in order to produce a report. It is also possible to perform some minor maintenance tasks like deciding on questions that are very difficult or answers that seems to be confusing.

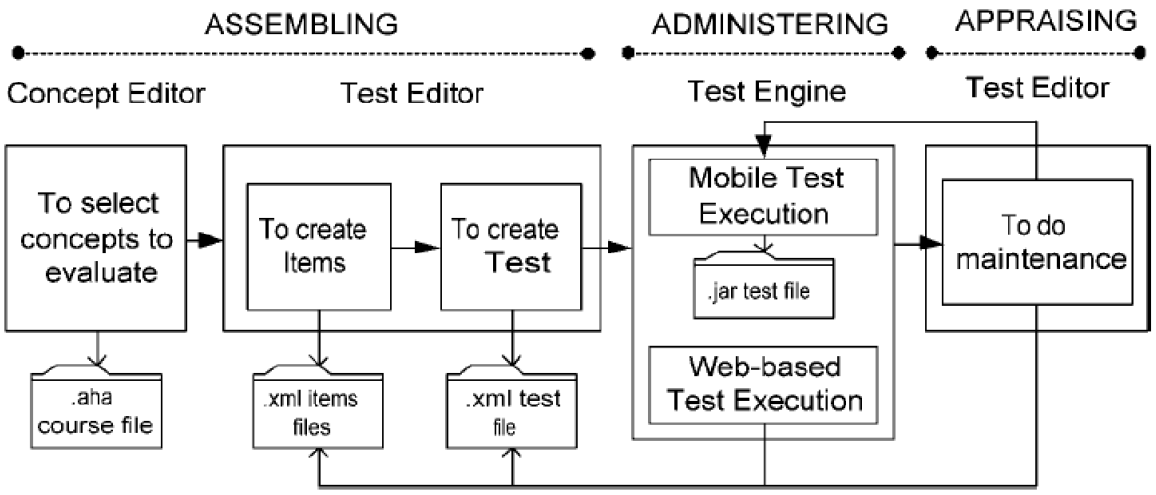

Figure 2.3: Triple-A Assessment Framework (Cristobal, Sebastian, and Paul, 2009)

2.3 Empirical Framework

This section of the research work discusses related works done by different research scholars under CBT system design, development, and deployment in order to establish a gap in knowledge in which this research work intends to fill. Fagbola, Adigun and Oke (2013) proposed a Computer-Based Testing (CBT) system for assembly, administering and appraising university online examination. The proposed system adopted the waterfall

model in the software prototype developmental life cycle. The proposed system was implemented using Microsoft SQL Server 2008, Macromedia Dreamweaver 8.0, and Microsoft Visual Studio 2012. Three key player (student, lecturer and administrator) interacted with the developed system, which had six (6) different active and functional modules. The system only administered multiple choice questions (MCQ) and structured questions, but recommended theory and video based question as a way forward.

Ajinaja (2017) also developed a computer based testing system for taking online examination using hybridization of waterfall model and the component based software engineering model. The system basically had three (3) key players that interacted with the designed application (students, teachers, and administrators). The proposed system delivered the e-assessment using multiple choice questions (MCQ) and structured questions format only, but recommended theory and video-based questions for e-assessment in the future. The application was developed using Hypertext preprocessor (PHP), JavaScript, Hypertext Markup Language (HTML), MySQL, Cascading Style Sheets (CSS) and XAMP Server.

Suleiman and Nachandiya (2018) proposed an online proctoring system for administering GST examination in Adamawa State University. The system adopted the Agile software Model. The design was implemented using PHP, MySQL, JavaScript, CSS, HTML and XAMP Server because it was developed to be a web-based application. The proposed system administered multiple choice questions (MCQ) that was randomly generated and different report can be generated immediately after the examination. The system considered authentication using username/password and MD5 encryption on its main security measures. Results also revealed that the CBT system is made up of six functional web pages for different activities.

Omotehinwa and Durojaye (2013) developed a computer based testing application that is web based with special interest in security and integrity of the output results. The system used the Unified Modeling Language (UML) for modeling and was implemented using PHP, HTML, and MySQL for the database structure design. The proposed system could administer questions such as Objectives, MCQ, True or False, and fill in the gaps questions format.

Ismail and Soye (2018) developed an improved computer based testing system by integrating biometric fingerprint and advanced encryption standard (AES) features to check for user's authentication and data integrity issues. The proposed developed based application was implemented using Java programming language and MySQL relational database management system. The system conceptual design was done using sequence and use case diagrams respectively.

Al-Saleem and Ullah (2014) considered security as a major threat with computer based testing application because of the vulnerabilities associated with online processes, and thereby proposed a novel hybridization of the conventional username/password verification structure with a palm-based biometric authentication process to the CBT application in order to proffer best security standard before, during and after the online examinations.

Adebayo and Aldulhamaid (2009) developed an electronic examination system especially for Nigerian Universities with special interest on securing the system against external attack and also to prevent alteration of the questions and results using the combination of biometric fingerprint technology, username/password and cryptographic technique. The proposed system was implemented using Java Applet, HTML, PHP, and MySQL. Web

browsers such as Internet Explorer, Netscape Navigator 9.0, Mozilla Firefox and Opera were used as different test bed for the application. Operating system such as Microsoft windows XP/7/Vista and Linux platforms were also used as the test environment for the proposed system. Results revealed the developed application to be efficient.

Wang et al. (2004) evaluated the acceptability of the Triple-A model commonly adopted and used in the development of computer based testing system for online test or examinations. Questionnaire was used in gathering user's responses about the model acceptability and Alpha Cronbach statistical techniques was used in the analysis in order to finding the level of satisfaction with the model structure and design. Findings revealed that the users were satisfied with the model.

Kuyoro *et al*. (2016) used the waterfall model development life cycle to design and develop a computer based testing application. The proposed application was implemented using web-based tools which comprises of Cascading Style Sheets (CSS), JavaScript, PHP, HTML, MySQL and WAMP Server as the deployment platform. The developed system was able to assemble, administer and appraise candidates for online examination, the administrator, lecturers, and students were the key players that interacted with the proposed CBT system.

Oluwole (2015) developed a desktop-based online examination system for university environment. The proposed application was implemented using Java programming language and MySQL database management system deployed over a wireless network environment for ease of connectivity.

2.4 Summary of the related works

This section summarizes the related work done under the empirical framework in a tabular format using the following criteria: author's name / year, purpose, method used and researcher's findings.

Table 2.1: Summary of the Related Works

S/N	Author/Year	Purpose	Method used	Findings
1	Fagbola, Adigun and Oke (2013)	Conducting Online examination	Waterfall model, Microsoft SQL Server 2008, Macromedia Dreamweaver 8.0, and Microsoft Visual Studio 2012	Just for conducting online examination using the Triple-A model only.
2	Ajinaja (2017)	Conducting Online examination	Waterfall model and component based software engineering model. PHP, JavaScript, HTML, MySQL, CSS and XAMP Server.	Just for conducting online examination using the Triple-A model only.
3	Suleiman and Nachandiya (2018)	Conducting Online examination	Agile software Model. PHP, MySQL, JavaScript, CSS, HTML and XAMP Server	Just for conducting online examination using the Triple-A model only.
4	Omotehinwa and Durojaye (2013)	Conducting Online examination	Unified Modelling Language (UML). PHP, HTML, and MySQL	For conducting online examination using the Triple-A model only. Emphasis on securing results.
5	Ismail and Soye (2018)	Securing Online examination	Biometric fingerprint and Advanced Encryption Standard (AES). Java programming language and MySQL. Sequence and Use case diagrams	For securing the conduct of online examination.
6	Al-Saleem and Ullah (2014)	Mitigating threat associated with computer based testing application for online examination	A hybridization of the conventional username/password verification structure with palm-based	For securing the conduct of online examination

			biometric authentication.	
7	Adebayo and Abdulhamid (2009)	Mitigating threat associated with computer based testing application for online examination and the integrity of questions and results	Biometric fingerprint technology, username/password and cryptographic technique. Java Applet, HTML, PHP, and MySQL.	For securing the conduct of online examination
8	Wang et al. (2004)	For evaluating the acceptability of the Triple-A model in CBT application.	Alpha Cronbach statistical tool	For evaluation of CBT in the conduct of online examination
9	Kuyoro *et al*. (2016)	Web-based online examination system	Waterfall model, Cascading Style Sheets (CSS), JavaScript, PHP, HTML, MySQL and WAMP Server	Just for conducting online examination using the Triple-A model only
10	Oluwole (2015)	Desktop-based online examination system	Java programming language and MySQL	Just for conducting online examination using the Triple-A model only

(Researcher, 2020)

CHAPTER THREE
METHODOLOGY

3.1 Data gathering techniques

This section critically describes the different data gathering technique used in this research work. The researcher combined both primary and secondary data collection techniques in order to have a holistic view of the study. The key informant interview, observation, published works and questionnaire techniques were used in this research work.

3.1.1 Key informant interview

The key informant interview method is a process of getting a professional opinion or interviewing an expert in order to gather very meaningful and precise facts about a particular key domain area as compared to interviewing a large population with little or no meaningful insights. The researchers conducted a structured interview sessions for five (5) administrators in-charge of Jamb CBT centres, three (3) professional software developers, two (2) network engineers, and two (2) Core IT expert with the sole aim of gaining special insights about CBT application design and development, network design, and the strengths and weaknesses associated with CBT application usage.

3.1.2 Observation method

In this section, the researchers observed some of the existing CBT applications readily available like the Moodle LMS, Jamb CBT Application, and few other CBT products developed by some researchers/developers currently being used by some institutions. We critically examined the following: the CBT Tripple-A methodology (Assembling, Administering, and Appraising), CBT deployment architecture, if it is desktop-based or

web-based, security considerations on authentication and database integrity for the questions using cryptographic techniques, type of assessment test (adaptive or linear), type of questions (Multiple Choice Questions (MCQ), fill in the gaps, objectives, scenario-based, video-based), enrollment of users on the CBT platform, formatting of questions, upload of single/bulk-based questions, speed of questions retrieval when using text-based questions and graphical/mathematical notations/scenario-based question, ability for the application to curb examination impersonators, and so on.

3.1.3 Published articles

Under this section the researchers downloaded and reviewed a lot of published scholarly articles based on CBT architecture, development and deployment from open journal access in order to critically assess the related work done on the study area and to justify the gap in knowledge established (See the summary of related work in Table 2.1).

3.1.4 Questionnaire

Under this section, the researchers gathered useful insight about the developed Smart CBT model's usability / acceptability using questionnaire responses from sixty (60) users. Table 4.13 showed the summary of user's responses from the questionnaire received, while Figures 4.32 – 4.42 showed the graphical analysis/interpretations from Table 4.13.

3.2 Software Methodology Overview

Software methodology can be referred to as a process life cycle for building a quality software product from start to finish. Generally, there are different types of software methodology with life cycle that varies from five to eight stages or phases as the case may be. This section of the research work critically examined software methodology such as waterfall model, prototyping model, spiral model, iterative model, incremental model, v-model and agile methodology. Their strengths and weaknesses were also properly examined in order to adopt a particular method that was most suitable for the proposed

intelligent agent service design and development for detecting impersonators in online examination environment.

3.2.1 Software methodology adopted

Haven critically examined the roles, functionalities, strengths and weaknesses of waterfall, prototyping, spiral, iterative, incremental, and v-model software development life cycles respectively, we arrived at a logical conclusion that they cannot be adopted as the chosen software development life cycle for the proposed system development based on their operational deficiencies and weaknesses; hence the agile model is chosen and adopted for usage based on its strengths.

The agile model performs tactic learning requirements and evolving solutions through the collective effort of self-organizing and cross-purposeful crews and their clients/end-users. The model sponsors adaptive design, evolutionary improvement, timely transfer, and persistent enhancement. It also inspires flexible rejoinders to modifications.

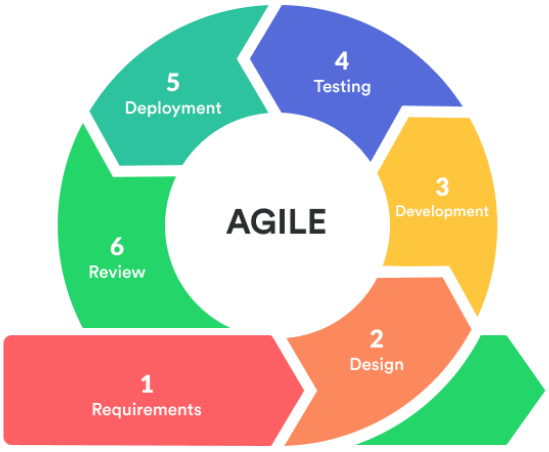

Figure 3.1: Agile life cycle model adopted

Figure 3.1 represents the diagram of the agile development life cycle with six (6) phases starting from requirements, design, development, testing, deployment and finally, the

review phase. All the activities in each of the phases and how the activities were carried out are explained in details under sections 3.3 – 3.8.

Table 3.1: Advantages and disadvantages of agile software methodology

Advantages	Disadvantages
▪ Agile model ensures clients fulfillment ▪ Agile model allows clients to give continual inputs and also have a face to face communication with the team member ▪ In agile model documentation of requirements is very precise, and it does not warrant presumption. ▪ Agile methodology is very dynamic; hence it can easily adjust with altering requirements. ▪ Agile methodology does not require committed time and determination since client requirements can change easily.	▪ Agile model could be very difficult in it comes to documenting large projects ▪ High expertise and knowledge is a major requirement from the agile team members ▪ In agile model, system design is not greatly stressed ▪ In agile model, clients can be easily confused if the results are not clear. ▪ Conducting test in the agile model could be sometimes difficult since it requires dedicated expertise

(Wikipedia, 2020)

3.3 Analysis of the existing CBT system

The existing CBT platform basically adopted the Triple-A ( ) model in conducting online examinations with other embedded features like creating users, reset password, upload questions, bulk upload, set timer for examination, user authentication, reporting, etc

This section carefully observed and analyzed six (6) existing CBT systems for conducting online test/quizzes/examination. Two (2) out of the six CBT systems are industry enterprise solutions namely; Moodle and JAMB CBT Application, while the other four (4) are CBT applications designed and developed based on research problems.

The twelve CBT systems were analyzed using the following criteria: web architecture, deployment mode, process flow, type of CBT, security measures, mode of preparing

questions, method of appraisal/assessment, results presentation, result checking, monitoring system, web browser compatibility, web browser security, user interface/user experience design and event notification.

Moodle Online Examination System (MOES)

Moodle is an abbreviation for "Modular Object Oriented Dynamic Learning Environment." It is an e-educational platform that delivers customized learning environments for students generally. Institutions can use the Moodle platform to create lessons, manage courses, and interact with both teachers/instructors/lecturers and students. Students can use Moodle to submit assignments, take quizzes/tests/examinations, and also relate with their mates based on its deployment (Internet or intranet).

Moodle is an open source enterprise application that has evolved since 1999 as a learning management software/platform. Table 3.2 shows detailed observation/findings done on Moodle LMS, while Figure 3.2 depicts the screenshot of a Moodle user's dashboard.

Table 3.2: Analysis done on the Moodle application

S/N	Criteria	Observation/findings
1	Web architecture	3-Tier over the Internet/intranet
2	Deployment mode	Web-based
3	Data Process flow	3 key players process flow of information
4	Type of CBT	Linear
5	Security measures	Username/password authentication
6	Mode of preparing question	Questions are usually preformatted before upload
7	Types of questions	MCQ and Structured questions format
8	Method of appraisal/assessment	Triple-A Method (assembling, administering and appraising)
9	Results presentation	The CBT system can generate and present results in different formats
10	Result checking	Result checking tab can be enabled to allow students easily their check results by clicking of the view result link or disabled as the case may be.

11	Monitoring system	Manual system of checking/monitoring students against impersonation using human invigilators and CCTV system
12	Web browser compatibility	Available browser compatibility; Internet Explorer+, Mozilla Firefox, Google chrome, etc
13	Web browser security	Not available. Students can easily open new tabs in search of information
14	User Interface (UI)/User Experience (UX) design	Easy to use user interfaces and higher responsive user experience design
15	Event notification	Multiple notification system

(Field work, 2020)

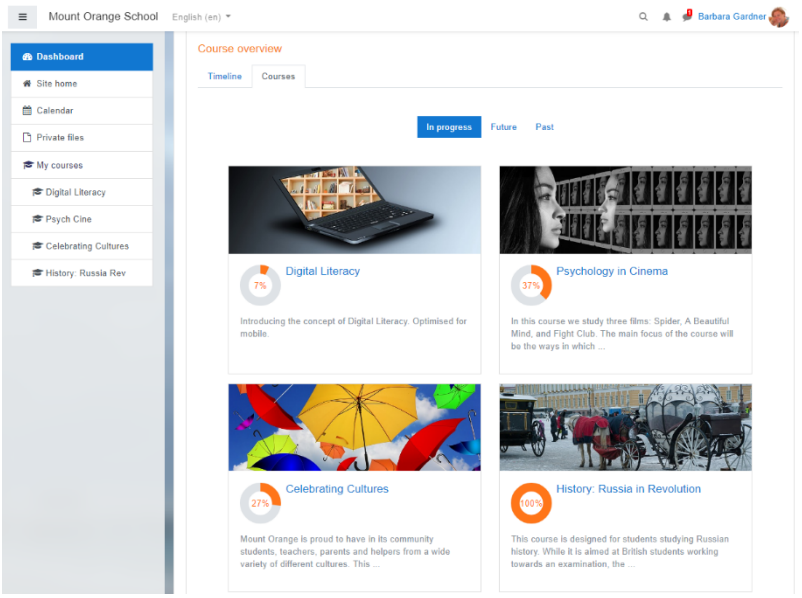

Figure 3.2: Screenshot of a Moodle user dashboard (Moodle LMS, 2020)

JAMB CBT enterprise application

The JAMB CBT Application is also an enterprise solution owned by the Joint Admission and Matriculation Board (JAMB) examination body in Nigeria charged with the responsibility for conducting entrance examination into Nigerian Universities.

Figure 3.3: JAMB CBT Centre

Table 3.3: Analysis done on the JAMB CBT Application

S/N	Criteria	Observation/findings
1	Web architecture	3-Tier over the intranet
2	Deployment mode	Desktop-based application
3	Data Process flow	2 key players process flow of information
4	Type of CBT	Linear
5	Security measures	Username/password authentication
6	Mode of preparing question	Questions are usually preformatted before upload
7	Types of questions	MCQ and Structured questions format
8	Method of appraisal/assessment	Triple-A Method (assembling, administering and appraising)
9	Results presentation	The CBT system can generate and present results in different formats
10	Result checking	Students check results after the examination on a different platform
11	Monitoring system	Manual system of checking/monitoring students against impersonation using human invigilators and CCTV system
12	Web browser compatibility	Not compatible. Application cannot execute on other browsers.
13	Web browser security	Customized secured browser to prevent students from opening new tabs in searching for information
14	User Interface (UI)/User Experience (UX) design	Easy to use user interfaces
15	Event notification	Simple pop-up notification system

(Field work, 2020)

Figure 3.4: JAMB candidate examination interface

CBT Systems developed based on research problems

Tables 3.4 – 3.7 shows the detailed analysis done on different CBT applications designed and developed by different research scholars.

Table 3.4: CBT system

S/N	Criteria	Observation/findings
1	Web architecture	3-Tier over the intranet
2	Deployment mode	Web-based
3	Data Process flow	3 key players process flow of information
4	Type of CBT	Linear
5	Security measures	Username/password authentication
6	Mode of preparing question	Questions are usually preformatted before upload
7	Types of questions	MCQ and Structured questions format
8	Method of appraisal/assessment	Triple-A Method (assembling, administering and appraising)
9	Results presentation	The CBT system can generate and present results in different formats
10	Result checking	Students can easily check results by clicking of the view result link
11	Monitoring system	Manual system of checking/monitoring students against impersonation using human invigilators and CCTV system
12	Web browser compatibility	Internet Explorer+, Mozilla Firefox, Google chrome
13	Web browser security	Not available. Students can easily open new tabs in search of information
14	User Interface (UI)/User Experience (UX) design	Easy to use and flexible user interfaces and simple structured user experience
15	Event notification	Simple pop-up notification system

(**Fagbola, Adigun and Oke, 2013)**

Table 3.5: CBT system

S/N	Criteria	Observation/findings
1	Web architecture	3-Tier over the intranet
2	Deployment mode	Web-based over LAN
3	Data Process flow	3 key players process flow of information
4	Type of CBT	Linear
5	Security measures	Login for students using registration number, then the full names of the student is displayed on the student's dashboard
6	Mode of preparing question	Questions are usually preformatted before upload
7	Types of questions	MCQ and Structured questions format
8	Method of appraisal/assessment	Triple-A Method (assembling, administering and appraising)
9	Results presentation	The CBT system can generate and present results in different formats
10	Result checking	Students can easily check results by clicking of the view result link
11	Monitoring system	Manual system of checking/monitoring students against impersonation using human invigilators and CCTV system
12	Web browser compatibility	Mozilla Firefox, Google chrome, etc
13	Web browser security	Not available. Students can easily open new tabs in search of information
14	User Interface (UI)/User Experience (UX) design	Simple and flexible UI/UX
15	Event notification	Result notification system

(**Ajinaja, 2017)**

Table 3.6: CBT system

S/N	Criteria	Observation/findings
1	Web architecture	3-Tier over the LAN
2	Deployment mode	Web-based
3	Data Process flow	3 key players process flow of information
4	Type of CBT	Linear
5	Security measures	Encrypted user login using MD5 and user privileges
6	Mode of preparing question	Questions are usually preformatted before upload
7	Types of questions	MCQ and fill in the blank spaces
8	Method of appraisal/assessment	Triple-A Method (assembling, administering and appraising)
9	Results presentation	The CBT system can generate and present results in different formats
10	Result checking	Students can easily check results by clicking of the view result link
11	Monitoring system	Manual system of checking/monitoring students against impersonation using human invigilators and CCTV system
12	Web browser compatibility	Compatible with different web browsers
13	Web browser security	Not available. Students can easily open new tabs in search of information
14	User Interface (UI)/User Experience (UX) design	Easy to use UI/UX
15	Event notification	Simple screen pop-up notification system

(Suleiman and Nachandiya, 2018)

Table 3.7: CBT system

S/N	Criteria	Observation/findings
1	Web architecture	2-Tier over local area network (LAN)
2	Deployment mode	Desktop-based application
3	Data Process flow	3 key players process flow of information
4	Type of CBT	Linear
5	Security measures	Username/password and biometric authentication system, advanced encryption standard (AES)
6	Mode of preparing question	Questions are usually preformatted before upload
7	Types of questions	MCQ and structured questions format
8	Method of appraisal/assessment	Triple-A Method (assembling, administering and appraising)
9	Results presentation	The CBT system can generate and present results in different formats
10	Result checking	Students can easily check results by clicking of the view result link
11	Monitoring system	Manual system of checking/monitoring students against impersonation using human invigilators and CCTV system
12	Web browser compatibility	Mozilla Firefox, Google chrome, etc
13	Web browser security	Not available. Students can easily open new tabs in search of information
14	User Interface (UI)/User Experience (UX) design	Easy to use
15	Event notification	Simple event pop-ups

(**Ismail and Soye, 2018)**

3.3.1 Weaknesses of the existing system

From the observation and analysis done as described in Tables 3.1 – 3.7; the major weaknesses with the existing system is in the area of security, which can be further explained as follows:

(a) Most of the CBT systems uses simple login authentication system which can be easily hacked to gain access.

(b) Most of the CBT system does not have the features to secure the database using cryptographic mechanisms

(c) The entire CBT system does not have a mechanism integrated to the CBT application that can check for impersonators without human intervention or the usage of CCTV cameras which are very passive security measures.

3.4 The proposed System

The proposed system is an improvement upon the existing online examination system (OES) or CBT application that simply uses the Triple-A model for assembling, appraising and assessment of student with some level of security measures like username/password/biometric authentications. It is a fusion or integration of an intelligent agent services using the K-NN classification machine learning algorithm to the existing CBT system for detection of likely suspected cases of impersonators during an online examination.

The intelligent agent services works within the same domain area or environment of the CBT; in order to carry out its specific function of detecting impersonators. The proposed system was modeled and analyzed using the notations as follows:

(i) IASOES = {IAS, OES, S}

Where:

IASOES = Intelligent Agent Service Online Examination System

IAS = Intelligent Agent Services communicating within the OES environment in carrying out its specific functions (security and detection of impersonators)

OES = Online Examination System for conducting examination

S = A set of processing nodes in which the intelligent agent services can be executed

(ii) The set of processing nodes is denoted as:

$S = \{s_1, s_2, s_3, \ldots, s_n\}$

Each node s_1 can provide an operating environment for the intelligent agent

(iii) The intelligent agent services (IAS) = {assessor_Agent, classifier_Agent}

The IAS is designed to profile and analyze each student user with a tendency to impersonate during the online examination. The IAS analysis is done using the Euclidean Distance (ED) type technique within the K-NN classification machine learning model.

The ED is a type nearest neighbor technique that can be used by the K-NN to compute for the nearest distance between two (2) points in an n-dimensional space. The designed model is able to use this technique to properly classify a scenario or condition if a particular student user is impersonating or not whereby point-1 is defined as "Impersonation" with certain value and point-2 is defined as "Not impersonating" also with certain value. The ED formula is defined by:

$$ED = \sqrt{[(a_1 - a_2)^2 + (b_1 - b_2)^2 + \ldots + (z_1 - z_2)^2]}$$

It should be noted that two (2) main parameters were taken into consideration for designing the classifier in order to ensure its effectiveness and efficiency. The parameters are as follows:

(i) 1st Parameter = the accuracy of the answered question asked by the intelligent agent assessor.

(ii) 2nd Parameter = the time taken to respond or answer a particular question is also noted by the intelligent agent assessor.

The model was trained and tested using a dataset with 3,083 records. The dataset was divided into two; 80% (2,466 records) of the dataset was used for training the model

during the training phase, while 20% (617 records) was used for testing the model during the testing phase (See Appendixes A and B).

This was done to enable the developed model predict future unseen data accurately; otherwise the designed and developed model will be over-fitted and unable to predict properly and accurately thereby increasing the false alarm rates.

3.4.1 Advantages of the Proposed System

The major advantages of the proposed system are as follows:

(i) The integration multiple of security measures in the areas of secured login authentication using 256-bit Salted Secure Hash Algorithm, generation of randomized examination questions using Merssene Twister Algorithm (MTA), secured examination questions stored in the database using the 128-Bit Advanced Encryption Standard (AES), and finally, secured unique examination codes using Message Digest 5 Algorithm (MD5).

(ii) Intelligent gathering, monitoring, detection and classification of any suspected case of impersonation during the online examination.

3.4.2 User and system requirements

This section discusses both the requirements expected from the application users and the requirements expected from the application developed; hence, we will look at the two requirements in the following sub-sections critically.

User requirements

The sub-section discusses the necessary requirements from the users in order to design and develop an effective and efficient application that will meet the users' expected and acceptable standard. The user's requirements were collected from three (3) key players which are as follows:

(a) Administrator

(b) Students

(c) Agent

It should be noted that these requirements will actually determine the roles / functions the users will play while using the developed Smart CBT application.

(a) The administrator's requirements

EPICS	REQUIREMENTS	USER STORY [Admin]
System Access	Login	As an Admin I would like to login to the System
System Access	Logout	As an Admin I would like to logout of the System
System Access	Reset Password	As an Admin I would like to Reset password
Registration	Search, edit and update students information	As an Admin I would like to search, edit and update student information
Registration	Register user for examination	As an Admin I would like to register students for examination
Registration	Add / Register students for a particular examination	As an Admin I would like to Register students for a particular examination
Registration	Create default passwords for all students	As an Admin I would like to create default passwords for students
Examination	Upload matching answers	As an Admin I would like to upload matching answers
Examination	Start and end examination based on set timer	As an Admin I would like to end examination based on a set timer
Examination	Broadcast short message to all students during examination	As an Admin I would like to broadcast short messages to students during exami
Examination	Set timer for each course	As an Admin I would like to set timer for each course
Software Agent	Start, Pause, Resume and Stop the Software Agent	As an Admin I would like to pause,resume and stop software Agent
Report Generation	Download students exam grade / reports in Excel / Word /PDF formats	As an Admin I would like to export student records in excel/pdf/word
Report Generation	Print student grade / reports in Excel / Word /PDF formats	As an Admin I would like to print student records in excel/pdf/word
Report Generation	View student result per course taken	As an Admin I would like to view student result per course taken

Figure 3.5: Admin requirement gathering (Researcher, 2020)

(b) The student's requirements

EPICS	REQUIREMENTS	USER STORY [Student]
System Access	Login	As a Student I would like to login to the system
System Access	Log out	As a Student I would like to logout of the system
Examination	Take Examination	As a Student I would like to take Examination
Report Generation	View result	As a Student I would like to view Result

Figure 3.6: Student user requirement gathering (Researcher, 2020)

(c) The agent's requirements

EPICS	REQUIREMENTS	USER STORY [Software Agent]
state life cycle	[Start] [Pause] [Resume] [Stop]	The software agent should be designed to operate in a four state life cycle as it relates with its environment which is the CBT application.
System Access	blocks or freezes the running CBT application	The software agent blocks or freezes the running CBT application which is its environment and superimpose it another question screen, in order to prevent the examinee or students from using the CBT application at that particular point in time, and until the student answers the agent's question before it will give control to the CBT application to be active again for the students usage
Detection	randomly select questions built from that particular students profile	The software agent will randomly select questions built from that particular students profile using an efficient randomized algorithm. Note: the agent post only one question at a time
Detection	takes cognizance of the response time and the correctness of the answer	The software agent carefully watches out for the user response to the question asked. It takes cognizance of the response time and the correctness of the answer given
Detection	correctly classify or cluster the examinee or student into "Normal", "Moderate Treat" or "High Treat or Risk"	The agent should be able to reason correctly per question asked in order to correctly classify or cluster the examinee or student into "Normal", "Moderate Treat" or "High Treat or Risk" based on the usage of supervised or unsupervised machine learning algorithm. Note the agent sets a particular threshold value for the various classification
Detection	Give its final VERDICT	The agent should ask the examinee questions at certain interval before concluding or given its final VERDICT
Data Entry	Commit all entries or activities noticed to a Log file	The agent should send or commit all entries or activities noticed to a Log file
Notification	send quick feedback message to the administrator	The agent should be able to send quick feedback message to the administrator in-charge about the suspected case of impersonation.
System Access	logout the examinee or student from the CBT application	The agent should automatically logout the examinee or student from the CBT application immediately it detects the risk factor is high

Figure 3.7: Agent's requirement gathering (Researcher, 2020)

Software Requirement Specification

The system requirements are the expected functionalities from the developed proposed system. The system requirements are of two categories:

(a) Functional requirements

(b) Non-functional requirements

The features of the two requirements are clearly itemized below:

(a) Functional requirements

(i) The developed Smart CBT application should be able to secure all login entries using 256-bit Salted Secure Hash Algorithm.

(ii) The developed Smart CBT application should be able to generate randomized examination questions Merssene Twister Algorithm (MTA)

(iii) The developed Smart CBT application should be able to detect and properly classify the case of an impersonator during the online examination using the K – Nearest Neighbor (KNN) Algorithm.

(iv) The developed Smart CBT application should be able to secure all examination questions stored in the database using the 128-Bit Advanced Encryption Standard (AES) alongside with the IPSec.

(v) Each examinable course to be written has a unique code assigned to it. The code is used to pull the questions particular to only that examinable course from the pool of so many examinable courses in the database. The developed Smart CBT application should be able to secure the examination code by encrypting it using the Message Digest 5 Algorithm (MD5).

(vi) The developed Smart CBT application should be able to efficiently match questions with the correct answers and also compute the final grades correctly to prevent a case of false alarm.

(vii) The developed Smart CBT application should be able to perform mass auto-submission on expiration of the set time for the examination.

(viii) The developed Smart CBT application should be able to tick answered questions.

(ix) The developed Smart CBT application should be able to generate a log report of all events.

(b) Non-functional requirements

(i) The system should support the integration of both multiple questions and single questions generation in a single screen view or display.

(ii) The system should be able to display the total number of questions to be answered on either the left hand side of the screen display in a matrix format made up of rows and columns or at the bottom bar made up of rows only.

(iii) The system should be able to support Backup/Restore operations in case of unforeseen circumstances

(iv) The system should be users friendly in terms of navigational bars, scroll bar, etc with attractive graphical user interface (GUI) and effective users experience (UX) environment.

3.4.3 System Modeling

System modeling is the study of the use of different models in order to conceptualize and build systems in professional manner. This section discusses the system architecture, deployment diagram, use case diagram, sequence diagram, flowchart design and algorithms implemented.

System Architecture

System architecture can be seen as a conceptual model that describes the organization, performance, and interpretations of an information system. A system architecture can sometimes include system constituents and its sub-systems that will work together in order to implement the general system.

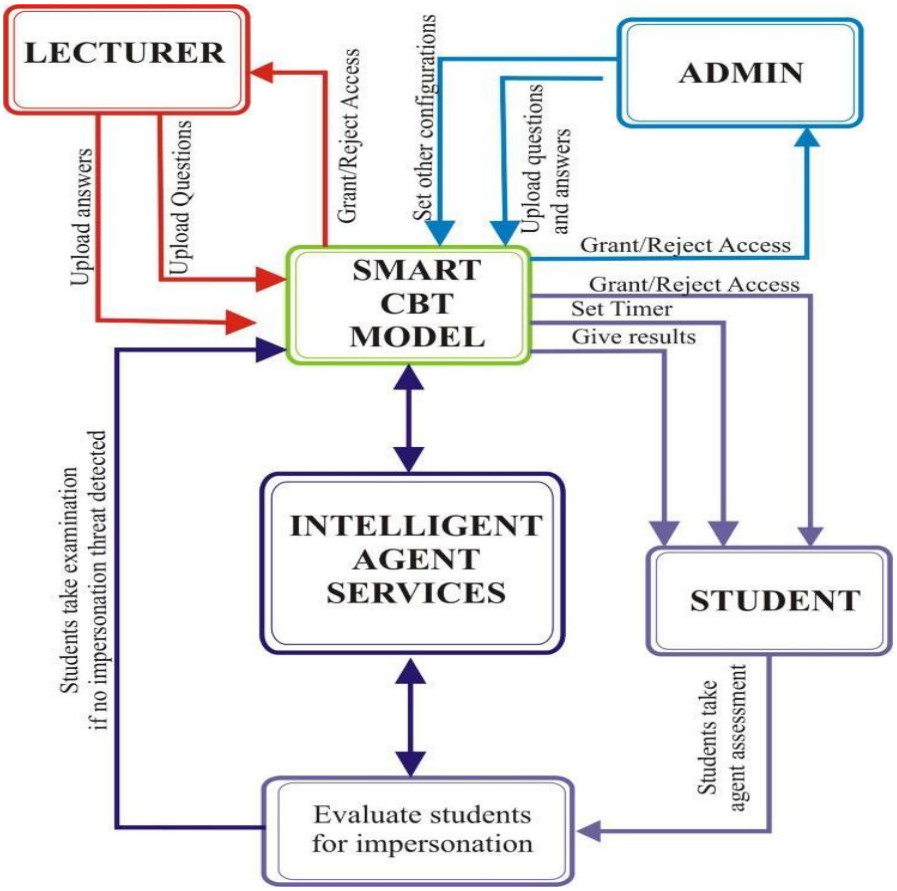

Figure 3.8: The Proposed System Architecture (Researcher, 2020)

Figure 3.8 depicts the proposed system architecture. It revealed the six (6) sub-systems namely; student module, lecturer module, admin module, CBT module, intelligent agent service module and the evaluation module. These sub-systems works together to achieve the desired goal.

Deployment diagram

A deployment diagram tries to model the physical deployment of an information system using nodes by revealing and describing the various type hardware components and their connections.

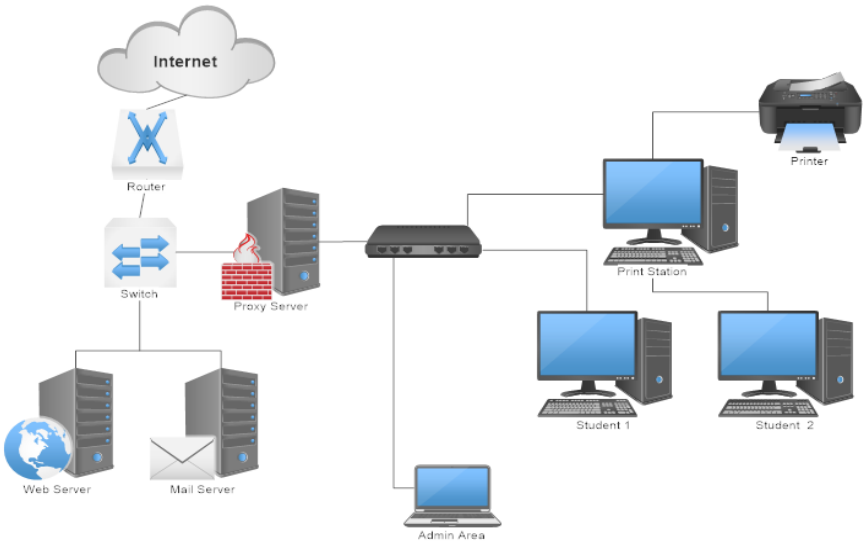

Figure 3.9: The Proposed System Deployment Diagram (Researcher, 2020)

Use case diagram

A use case diagram shows or represents user's interaction with a system. A use case diagram most times identifies the relationship between the different types of users of a system.

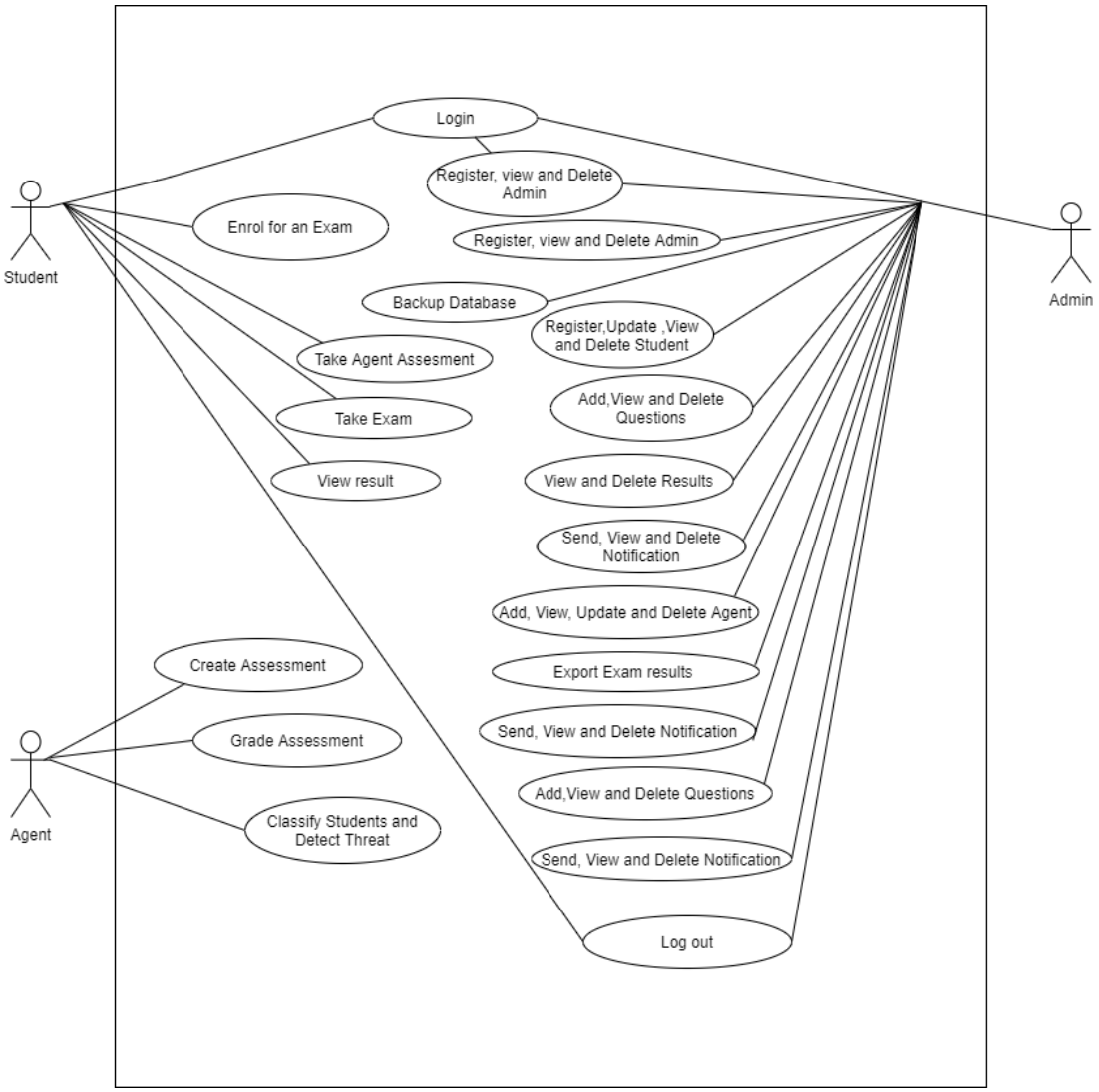

Figure 3.10: Use case diagram for the Proposed System (Researcher, 2020)

Figure 3.10 showed the use case diagram for the proposed system with the three (3) active key players (Admin, Student, and Agent) identified. The activities they can handle or execute on the propose information system and the relationship between the users are also revealed.

Sequence diagram

A sequence diagram displays object interactions prepared in time order. A sequence diagram displays parallel upright lines that represents the different objects or processes that interacts with each other, and the horizontal arrows represents the messages the objects exchange between themselves in the order of how the messages come about.

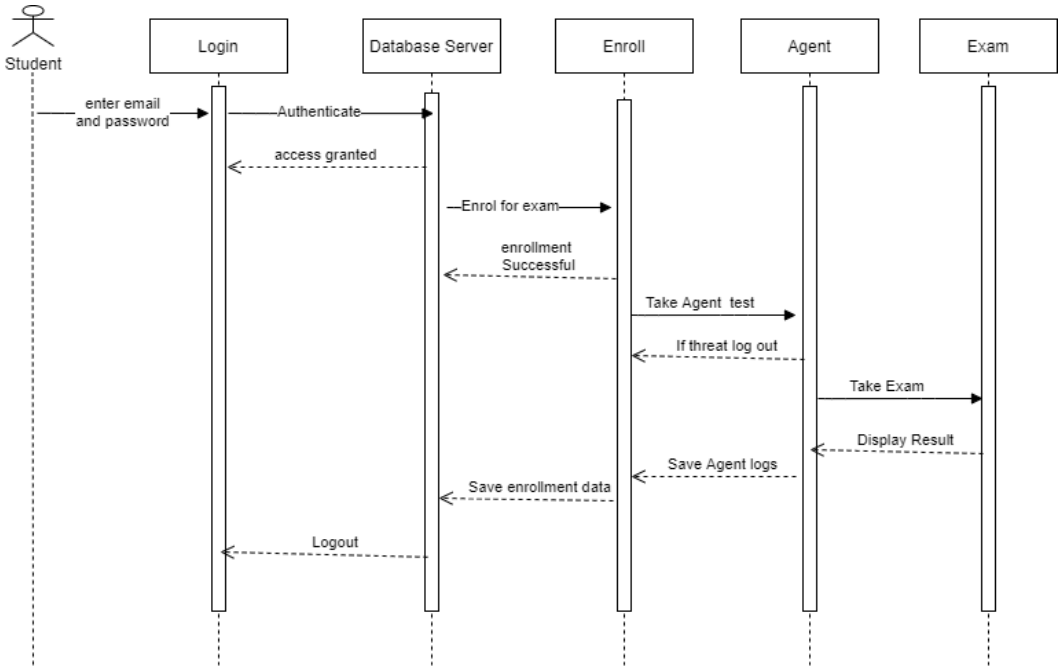

Figure 3.11: Sequence diagram of the propose system interaction (Researcher, 2020)

Figure 3.11 depicts a sequence diagram of how a student user can interact with the propose Smart CBT application from the login process through the examination process.

Flowchart design

A flowchart can simply be referred to as a diagrammatic representation of a step by step method of explaining a solution to a given problem or task.

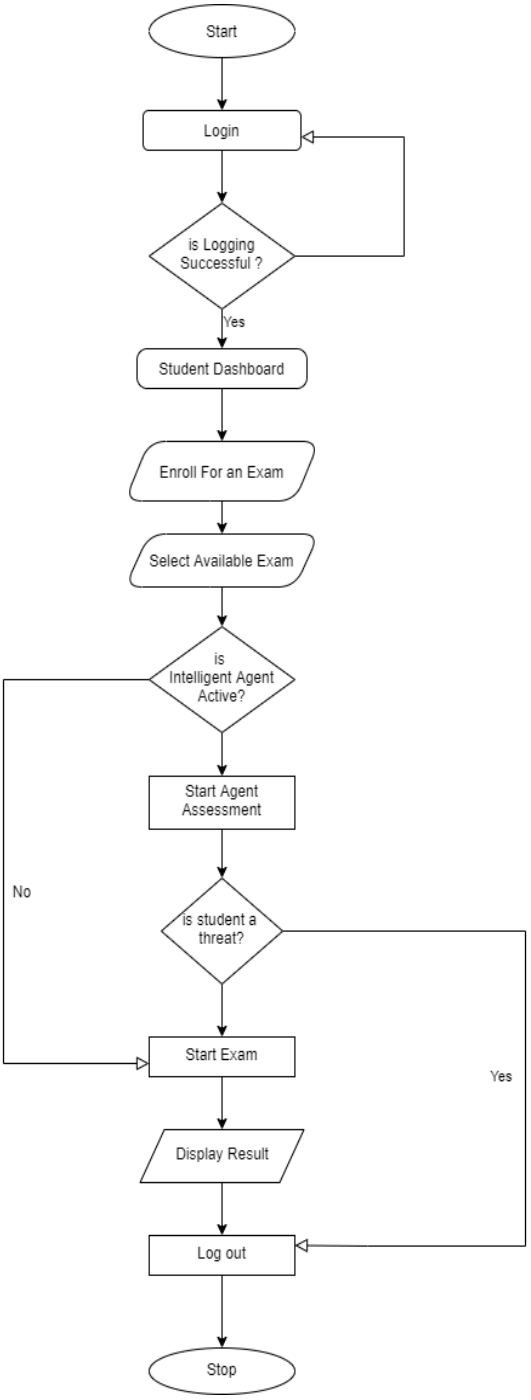

Figure 3.12: Process flowchart for taking online examination (Researcher, 2020)

Algorithm Implemented

Algorithm can be seen or defined as a finite arrangement of well-defined instructions to solve a given task or problem. This section deals with algorithms implemented in order to

ensure the developed software prototype is efficient based on time and complexity. The different algorithms used in the development of the proposed system are shown below.

(a) Merssene Twister Algorithm (MTA)

```
// Create a length n array to store the state of the generator
 int[0..n-1] MT
 int index := n+1
 const int lower_mask = (1 << r) - 1 // That is, the binary number of r 1's
 const int upper_mask = lowest w bits of (not lower_mask)
 // Initialize the generator from a seed
 function seed_mt(int seed) {
    index := n
    MT[0] := seed
    for i from 1 to (n - 1) { // loop over each element
       MT[i] := lowest w bits of (f * (MT[i-1] xor (MT[i-1] >> (w-2))) + i)
    }
 }
  // Extract a tempered value based on MT[index]
 // calling twist() every n numbers
 function extract_number() {
    if index >= n {
       if index > n {
         error "Generator was never seeded"
         // Alternatively, seed with constant value; 5489 is used in reference C code[52]
       }
       twist()
    }
     int y := MT[index]
    y := y xor ((y >> u) and d)
    y := y xor ((y << s) and b)
    y := y xor ((y << t) and c)
    y := y xor (y >> l)
   index := index + 1
    return lowest w bits of (y)
 }
  // Generate the next n values from the series x_i
 function twist() {
    for i from 0 to (n-1) {
       int x := (MT[i] and upper_mask)
                + (MT[(i+1) mod n] and lower_mask)
       int xA := x >> 1
       if (x mod 2) != 0 { // lowest bit of x is 1
          xA := xA xor a
       }
       MT[i] := MT[(i + m) mod n] xor xA
    }
    index := 0
 }
```

Figure 3.13: MT Algorithm for Randomization of Examination Questions (Researcher, 2020)

Figure 3.13 shows the Messene Twister Algorithm (MTA) which is a pseudorandom number generator (PRNG) used to randomly generate questions of the database. The MTA is an enhancement done on the Linear Congruential Generator algorithm.

(b) K–Nearest Neighbor (KNN) Algorithm

Step 1: For implementing the algorithm, we need a dataset. So during the first step of KNN, we must load the training as well as test data.
Step 2: Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.
Step 3: For each point in the test data do the following for (**Step 4 to step 7**)
Step 4: Compute the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. Though the most commonly used method to calculate distance is Euclidean.
Step 5: Based on the distance value, you can now sort them in ascending order.
Step 6: Next, it will choose the top K rows from the sorted array.
Step 7: Now, it will assign a class to the test point based on most frequent class of these rows.
Step 8: End

Figure 3.14: K-NN Algorithm (Researcher, 2020)

Figure 3.14 explains the KNN algorithm. The K-NN Algorithm is used for the classification of likely impersonators in the CBT environment. The KNN uses the Euclidean, Manhattan or Hamming distance Algorithm. But in this model the KNN was implemented using the Euclidean distance.

(c) Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES), also known by its original name is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

AES is a subset of the Rijndael block cipher developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen, who submitted a proposal to NIST during the AES selection process.

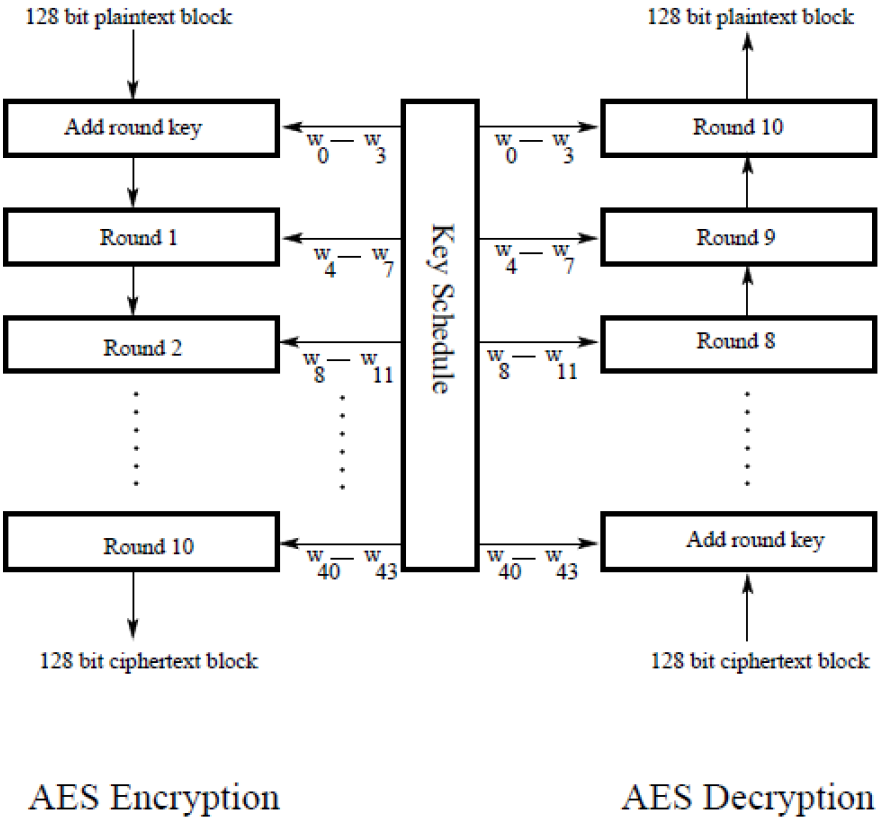

Figure 3.15: Advanced Encryption Standard (AES) 128 Bit Structure

Figure 3.15 depicts the overall structure of the AES. It is used to secure the examination questions stored in the database alongside with the IPSec Encapsulating Security Payload (ESP). The AES-128-CTR Encryption scheme was used with an Open SSL Encryption Method.

(d) Salted Secure Hash Algorithm 256-bit

Step 1: The user creates an account.

Step 2: Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive.

Step 3: When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database).

Step 4: If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials.

Step 5: Steps 3 and 4 are repeated every time someone tries to login to their account.

Figure 3.16: 256-bit Salted Secure Hash Algorithm

Figure 3.16 shows the 256-bit salted secure hash algorithm responsible for securing password, logging details and registration entries.

(e) Message-Digest Algorithm (MD5)

Step1: Append Padding Bits
Step 2: Append Length
Step 3: Initialize MD buffer
Step 4: Processing message in 16-word block

Figure 3.17: Message Digest Algorithm 5

Figure 3.17 shows the Message Digest Algorithm 5 (MD5), which is a cryptographic hash algorithm that can be used to create a 128-bit string value from an arbitrary length string. Although there has been insecurities identified with MD5, it is still widely used. MD5 is most commonly used to verify the integrity of files. However, it is also used in other security protocols and applications such as SSH, SSL, and IPSec. Some applications strengthen the MD5 algorithm by adding a salt value to the plaintext or by applying the hash function multiple times. The Message Digest algorithm 5, was created by Ronald Rivest. It is the most widely used of the MD family of hash algorithms. The MD5 was used in the smart CBT model to secure the examination code by encrypting it. Each

examination to be written has a unique code generate for that particular examination, and the code is used to pool the questions from the pool of question in the database; hence it needs to be secured using the MD5 algorithm.

3.4.4 Data Modeling

Data modeling is the process of building a data model for a new information system by applying induced proper procedures. Data modeling can be used to describe and examine data requirements desired to assist a business process within the possibility of consistent information systems. Data modeling methods and procedures are also used to model data in a typical, reliable, and foreseeable way in order to accomplish it as a resource.

Entity relationship diagram, class diagram, data flow diagram and data dictionary are the major data modeling methods discussed under this section.

Entity Relationship Diagram (ERD)

Entity relationship diagram also referred to as entity relationship model or E-R model tries to define the interconnected things of concern in a particular domain of information. The E-R model forms an abstract data model, that defines and systematically analysis a data or information structure which can be implemented in a relational database of an information system.

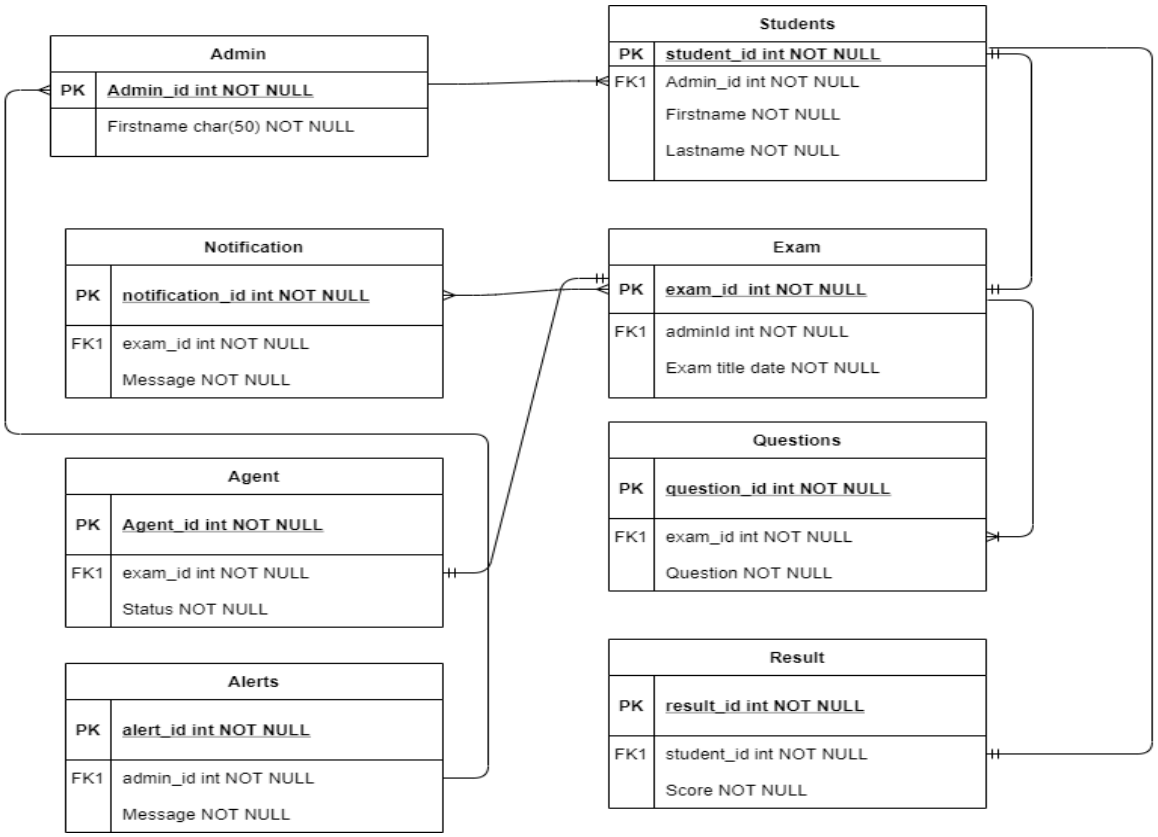

Figure 3.18: Entity relationship model for the propose system (Researcher, 2020)

Figure 3.18 depicts the E-R model or diagram of the propose system, which showed the interconnection and interrelationship amongst the admin, students, notification, examination, agent, questions, alert and results modules.

Class Diagram

A class diagram is a type of static structure diagram that defines the organization of a system by displaying the system's classes, qualities, processes, and the associations amongst objects. The class diagram is the focal building block of object-oriented modeling, and it is used for all-purpose abstract modeling of the structure of an application, and for comprehensive modeling by converting the models into code.

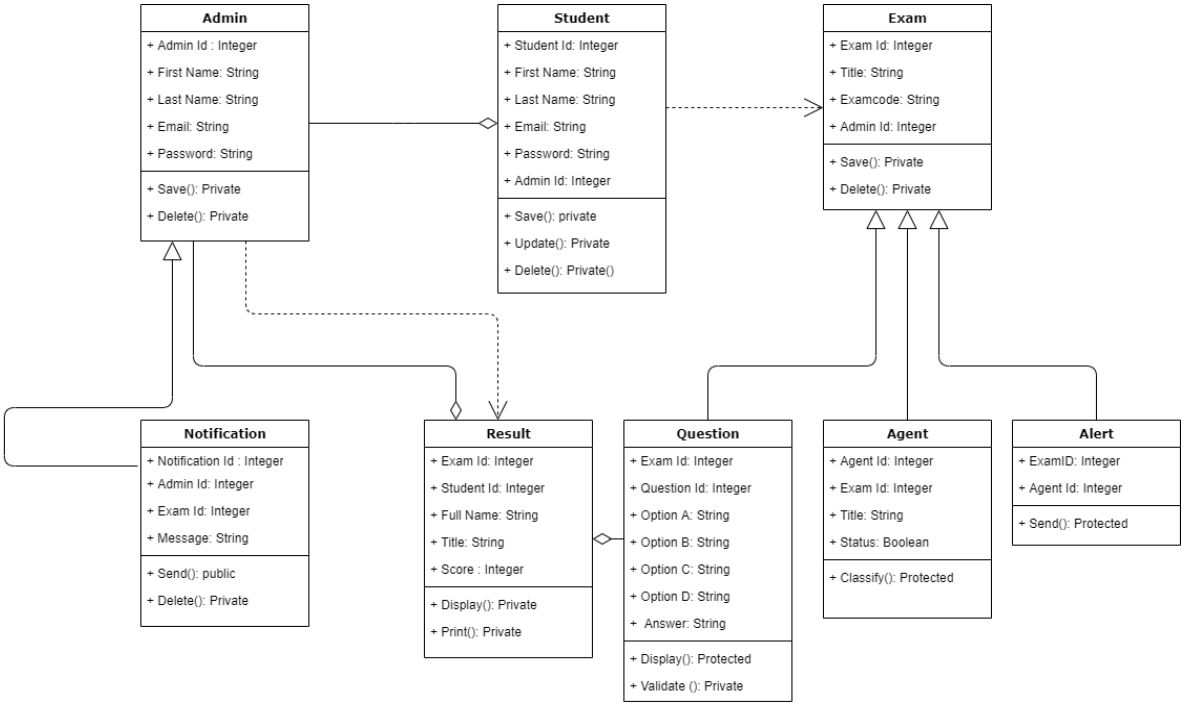

Figure 3.19: Class diagram of the propose system (Researcher, 2020)

Figure 3.19 depicts the class diagram of the propose system, which showed the relationship amongst the admin, students, notification, examination, agent, questions, alert and results modules.

Data Flow Diagram (DFD)

A data flow diagram is a method of expressive the movement of data through a process or an information system. The DFD offers information about the inputs and outputs of every entity and process of the information system, but it has no decision rules, loops or control flow.

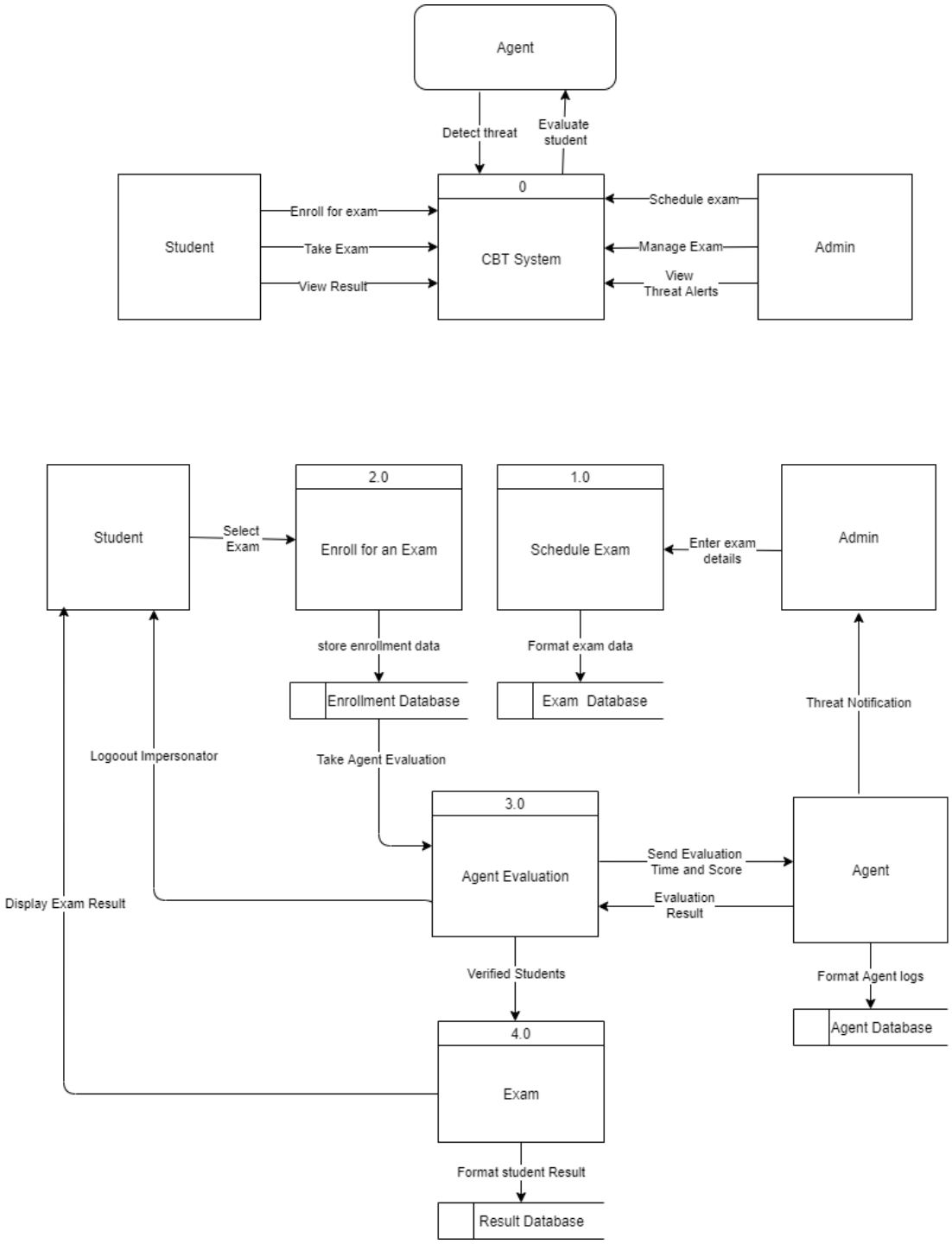

Figure 3.20: Proposes System Data Flow Diagram (Researcher, 2020)

Figure 3.20 depicts the data flow diagrams (DFD) of the propose system; the first DFD showed the movement of data amongst the admin, students, agent, and the CBT system,

while the second showed the movement of data amongst student, enroll for an exam, schedule an exam, admin, agent evaluation, agent, exam, and the results modules.

Data dictionary

A data dictionary which could also be referred to as a metadata repository is a central storehouse of information about data containing special details such as its meaning, relationship to other data, source, practice, and format. It can also be seen as a gathering of tables with metadata.

Tables 3.8 – 3.18 described the different tables / data dictionaries used in the proposed system.

Table 3.8: Admin data dictionary

Field Name	Data Type	Field Size	Description	Example
Admin ID	Integer(PK)	10	Unique Integer ID for all Admin	12345
First Name	VarChar	30	First Name for Admin	Joy
Last Name	VarChar	30	Last Name for Admin	Peter
Phone Number	Integer	15	Phone No. for Admin	08136458901
Email	VarChar	40	Email Address for Admin	ify@gmail.com
Password	VarChar	100	Password for Admin	Ldsldkl322$^&ue
Picture	Byte	200	Picture for Admin	Photo
Status	VarChar	10	Status for Admin	Active/inactive
Salt	VarChar	200	Salt for Admin password	%&889390393893839
Date	DateTime	20	Date of Record Entry	01/01/2020:1:35 PM

(Researcher, 2020)

Table 3.9: Agent data dictionary

Field Name	Data Type	Field Size	Description	Example
ID	Integer(PK)	10	Unique Integer ID for all Agent	12345
Exam ID	Integer (FK)	10	Unique Integer ID for all Exam	12345
Exam Title	VarChar	50	Title of Exam	English,Maths
Agent Status	VarChar	10	Status for Agent	Active
Exam Status	VarChar	10	Status for Exam	Pending

(Researcher, 2020)

Table 3.10: Alert data dictionary

Field Name	Data Type	Field Size	Description	Example
ID	Integer(PK)	10	Unique Integer ID for all Alert	12345
Exam ID	Integer (FK)	10	Unique Integer ID for all Exam	12345
Exam	VarChar	50	Title of Exam	English
Full Name	VarChar	30	Name of Student	James
Offense	VarChar	20	Offense of the students	Impersonation
Time	DateTime	20	Time of the Offense	12:00pm
Status	VarChar	10	Status for Exam	In Progress

(Researcher, 2020)

Table 3.11: Examination data dictionary

Field Name	Data Type	Field Size	Description	Example
Exam ID	Integer(PK)	10	Unique Integer ID for all Exam	12345
Admin ID	Integer(FK)	10	Unique Integer ID for all Admin	12345
Exam Title	VarChar	50	Title of Exam	Maths
Exam Date Time	DateTime	20	Date and time for Exam	1/0/20 1:20pm
Total Question	Integer	10	Number of Question	5,80,100
Marks per right answer	Integer	10	Mark for correct answers	1,2,3
Marks per wrong answer	Integer	10	Mark for wrong answers	-1,-2,0
Exam Status	VarChar	20	Status of Exam	Pending
Exam Code	VarChar	50	Unique ID for Exam	%6g2js42888
Date Created	DateTime	20	Exam Creation Date	1/0/20 1:20pm

(Researcher, 2020)

Table 3.12: Question data dictionary

Field Name	Data Type	Field Size	Description	Example
Question ID	Integer(PK)	10	Unique Integer ID for all Question	12345
Exam ID	Integer(FK)	10	Unique Integer ID for all Exam	12345
Question Title	VarChar	100	Exam Question	What is a noun?
Answer Option	VarChar	10	Correct Option for Exam Question	A,B,C,D
Option 1	VarChar	100	Option for Exam Question	A,B,C,D
Option 2	VarChar	100	Option for Exam Question	A,B,C,D
Option 3	VarChar	100	Option for Exam Question	A,B,C,D
Option 4	VarChar	100	Option for Exam Question	A,B,C,D

(Researcher, 2020)

Table 3.13: Result data dictionary

Field Name	Data Type	Field Size	Description	Example
ID	Integer(PK)	10	Unique Integer ID for all Result	12345
Student ID	Integer(FK)	10	Unique Integer ID for all Student	12345
Exam ID	Integer(FK)	10	Unique Integer ID for all Exam	12345

Full Name	VarChar	50	Full Name For Student	Samuel
Questions	Integer	10	Number of Question answered	50,60
Mark Obtained	Integer	10	Total Marks Obtained	1,2,4,10
Date	Date Time	20	Date and time of the Exam	01/01/2020 1:45 PM

(Researcher, 2020)

Table 3.14: Sent notification data dictionary

Field Name	Data Type	Field Size	Description	Example
NID	Integer(PK)	10	Unique Integer ID for all Notice	12345
Exam ID	Integer(FK)	10	Unique Integer ID for all Exam	12345
Admin ID	Integer(FK)	10	Unique Integer ID for all Admin	12345
Title	VarChar	50	Notification Title	Notice
Message	VarChar	200	Notification Message	Time's up
Status	VarChar	20	Message Status	Sent
Date	Date Time	10	Date and time of the Notice	01/01/2020 1:45 PM

(Researcher, 2020)

Table 3.15: User log data dictionary

Field Name	Data Type	Field Size	Description	Example
ID	Integer(PK)	10	Unique Integer ID for all Log	12345
Admin ID	Integer(FK)	10	Unique Integer ID for all Admin	12345
Full Name	VarChar	50	Full Name for the User	Tobi Adeola
Access Level	VarChar	50	Access Level for the User	Admin
Date Time	Date Time	20	Date and time of user login	01/01/2020 1:45 PM

(Researcher, 2020)

Table 3.16: Examination enrollment data dictionary

Field Name	Data Type	Field Size	Description	Example
Student Enrollment ID	Integer(PK)	10	Unique Integer ID for all Enrollment	12345
Student ID	Integer(FK)	10	Unique Integer ID for all Student	12345
Exam ID	Integer(FK)	10	Unique Integer ID for all Exam	12345
Exam Status	VarChar	20	Current Status of the Exam	pending

(Researcher, 2020)

Table 3.17: Student data dictionary

Field Name	Data Type	Field Size	Description	Example
Student ID	Integer(PK)	10	Unique Integer ID for all Student	1234
First Name	VarChar	30	First Name For Student	Sam
Middle Name	VarChar	30	Middle Name For Student	Joe
Last Name	VarChar	30	Last Name For Student	Abah
Email	VarChar	50	Email Address For Student	Flow@yahoo.com
Phone Number	Integer	30	Phone Number For Student	07033456678
Password	VarChar	100	Password For Student	D778383hd@#
Mother Maiden Name	VarChar	50	Mother Maiden Name for student	Mary
User Image	Byte	1000	User Image For Student	Photo
Favorite Food	VarChar	20	Favorite Food For Student	Rice
Favorite Sport	VarChar	20	Favorite Sport For Student	Football
Pet Name	VarChar	20	Pet Name For Student	Jack
State	VarChar	20	State For Student	Lagos
LGA	VarChar	20	LGA For Student	Shomolu
Date of Birth	VarChar	20	Date of birth For Student	01/01/1985
Town	VarChar	20	Town For Student	Abuja
Salt	VarChar	200	Salt For Student Password	%^#&&#*##()#
Date	Date Time	20	Date and Time of Registration	01/01/2020 1:45 PM

(Researcher, 2020)

Table 3.18: Agent logs data dictionary

Field Name	Data Type	Field Size	Description	Example
AGENT ID	Integer(PK)	10	Unique Integer ID for all Agent	12345
Student ID	Integer(FK)	10	Unique Integer ID for all Student	12345
Exam ID	Integer(FK)	10	Unique Integer ID for all Exam	12345
Score	Varchar	20	Score of the Student	pending
Time	Integer	20	Time Taken for the assessment	1,4,6,7
Classification	Varchar	20	Result of the prediction	Threat

Date Time	Date Time	20	Date and Time of Assessment	01/01/2020 1:45 PM

(Researcher, 2020)

3.4.5 System Implementation

Implementation is simply means the realization of a methodological design or software constituent through coding and software deployment.

Choice of Programming Languages

The choice of programming languages used in the development of the proposed system prototype model and the aspect they were used to build are listed and explained below.

(a) JavaScript was used for developing the exam platform and intelligent agent

(b) PHP was used for developing the server side functionalities of the entire system.

(c) MySQL was used for the database table design and query.

(d) Python programming language was used for the dataset preparation, preprocessing and algorithm evaluation

(e) HTML and CSS were used for the user interface design.

Development environment / tools used

The development environment and tools used in the development of the proposed system prototype model and the aspect they were used for are also listed and explained below:

(a) Visual Studio Code: Used for Writing the Entire Codebase of the Web Platform (HTML/CSS, PHP and JavaScript)

(b) Jupyter Notebook: for executing the Python language used in the dataset preparation, preprocessing and algorithm evaluation.

(c) XAMPP: XAMPP serves as the web server to host the entire platform on a local host.

3.4.6 Implementation procedures

This section discusses the different implementation procedures for Setting up Development Environment in Microsoft Windows Operating System

(a) How to setup Visual Studio Code on Windows

Firstly, download the Visual Studio Code installer for Windows. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). It will only take a minute.

Secondly, accept the agreement and click on next button to continue.

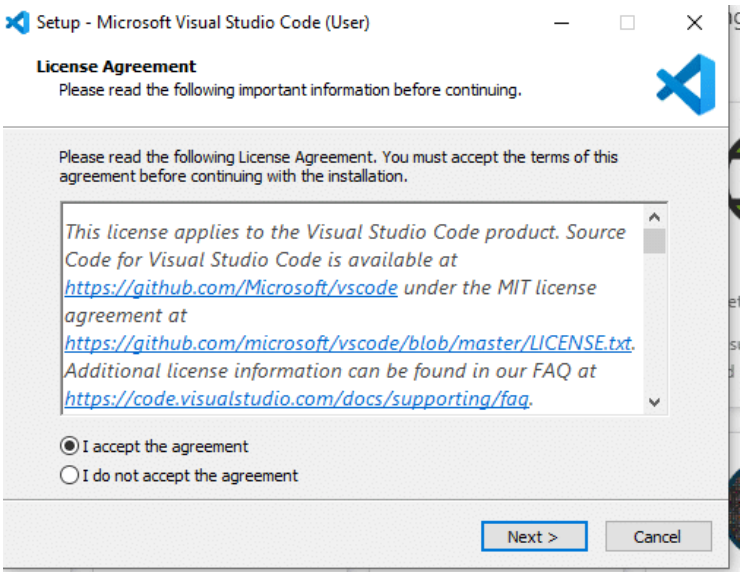

Figure 3.21: Microsoft Visual Studio Code License Agreement Interface (Microsoft Visual Studio, 2020)

Thirdly, click on "*create a desktop icon*" so that it can be accessed from desktop and click on Next button to continue.

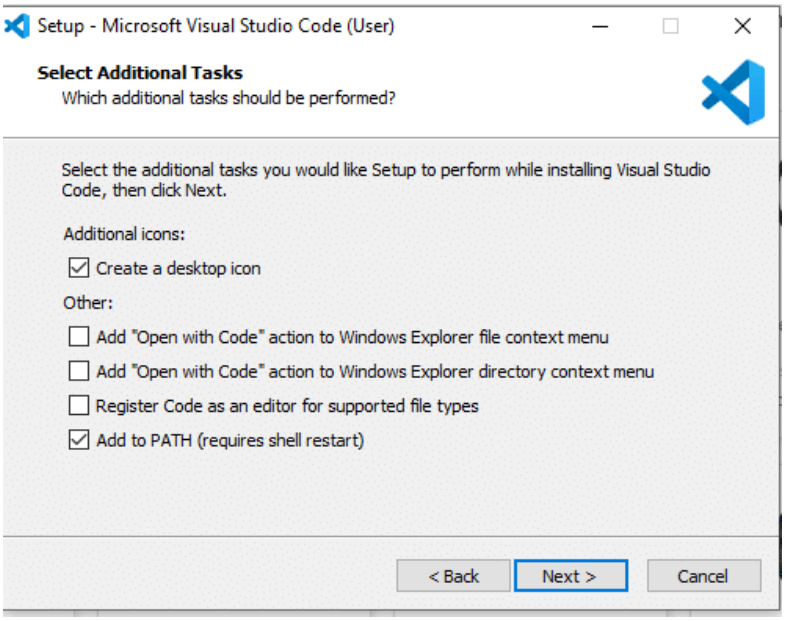

Figure 3.22: Microsoft Visual Studio Code Select Additional Task Interface (Microsoft Visual Studio, 2020)

After that, click on the install button.

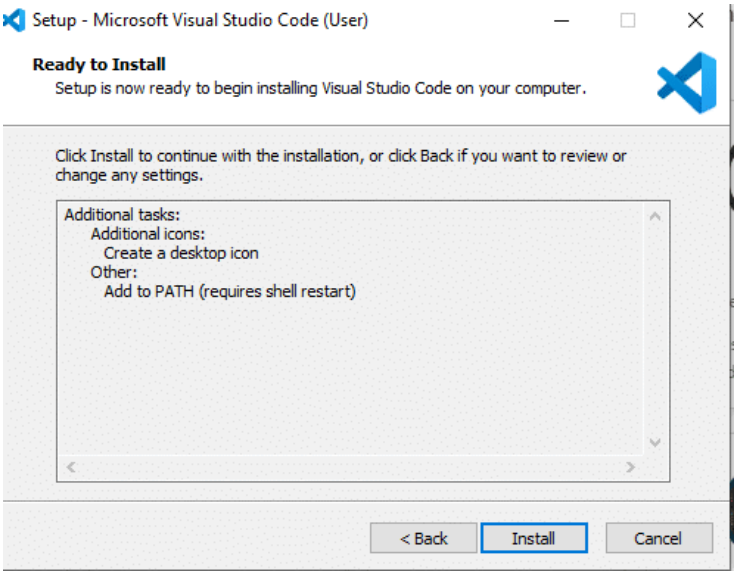

Figure 3.23: Microsoft Visual Studio Code Ready to Install Interface (Microsoft Visual Studio, 2020)

Finally, after installation completes, click on the finish button, and the visual studio code will get open.

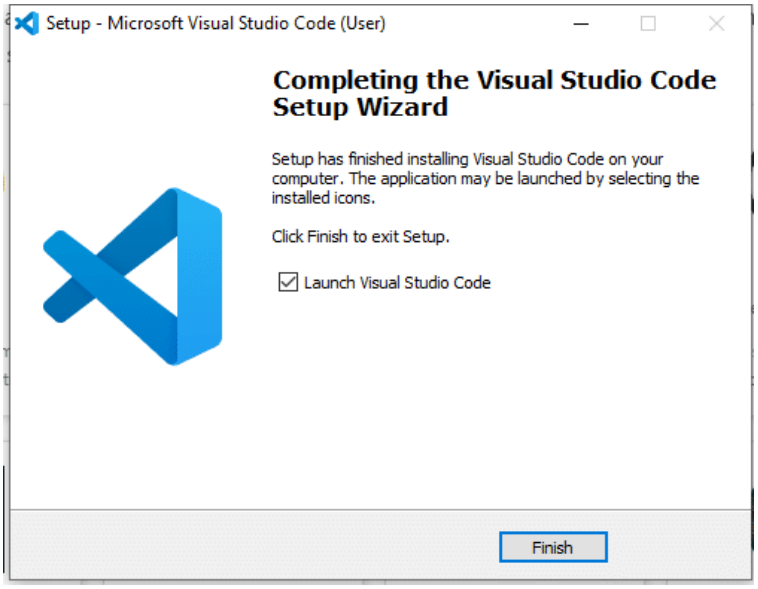

Figure 3.24: Completing the Installation Wizard Interface (Microsoft Visual Studio, 2020)

It should be noted that, by default, Microsoft Visual Studio Code is installed under C:\users\{username}\AppData\Local\Programs\Microsoft VS Code.

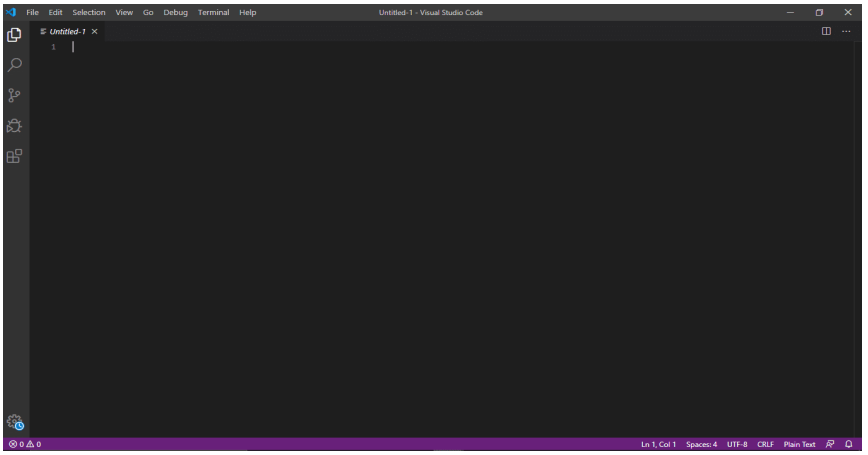

Figure 3.25: Microsoft Visual Studio Code Editor Interface (Microsoft Visual Studio, 2020)

After the successful installation, the next phase is to understand the various components of the User Interface of under Microsoft Visual Studio Code Editor. The Microsoft Visual Studio Code is a code editor at its core. It is like many other code editors, Microsoft VS

Code adopts a standard user interface and layout of an explorer on the left, showing all of the files and folders you have access to. Additionally, it has an editor on the right, showing the content of the files you have opened. Below are a few of the most critical components the Microsoft VS Code editor.

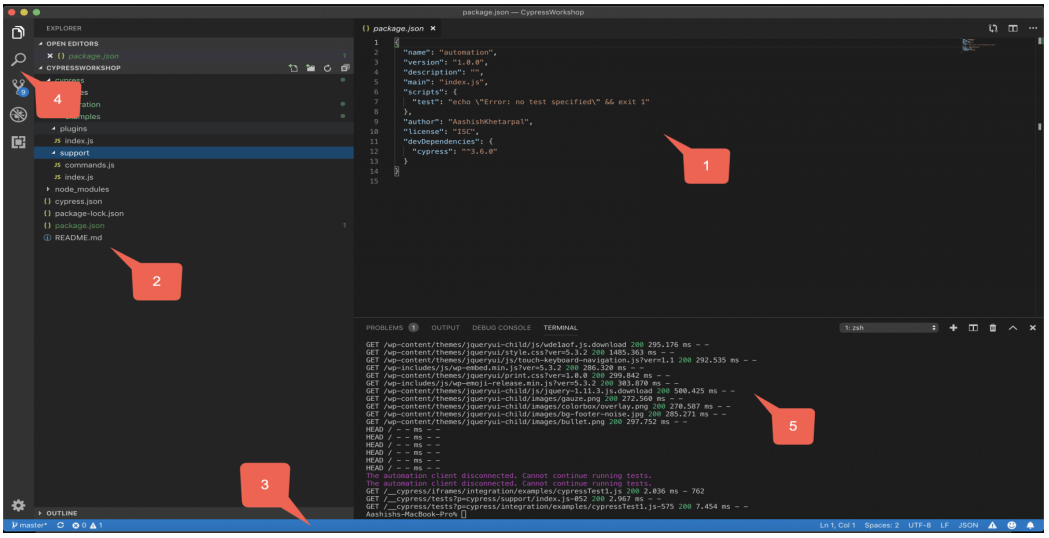

Figure 3.26: Microsoft Visual Studio Code Editor Environment (Microsoft Visual Studio, 2020)

The Microsoft Visual Studio Code comes with a straight forward and in-built outline that exploits the space delivered by the editor while leaving a sufficient room to browse. In addition, the Microsoft Visual Studio Code allows access to the full context of your folder or project. The user interface (UI) is divided into five areas, as highlighted in Figure 3.xx.

1. Editor – It is the main area to edit your files. You can open as many editors as possible side by side vertically or horizontally.
2. Sidebar – Contains different views like the Explorer to assist you while working on your project.
3. Status Bar – It contains the information about the opened project and the files you edit.

4. Activity Bar – It is located on the far left-hand side. It allows you to switch between views and also gives you an additional context-specific indicators like the number of outgoing changes when the Git is enabled.
5. Panels – It displays different panels below the editor region for output or debug information, errors, and warnings, or an integrated terminal. In addition, the panel can also move to the right for more vertical space.

(b) **How to install Jupiter Notebook in Windows**

The Jupyter Notebook is an open-source web application that simply allows someone to create and share documents that contains live programmable codes, calculations, visualizations, and descriptive text. It has varieties of functions, which includes data cleaning and transformation, machine learning, mathematical simulation, data visualization, statistical modeling, and so on.

Jupyter Notebook using Anaconda

Anaconda is an open-source software that contains Jupyter, spyder, etc which can be used for large data processing, data analytics, and weighty scientific computing. Anaconda works well for R and Python programming language. Spyder which is a sub-application of Anaconda that is also used with Python.

Installation procedures for Jupyter using Anaconda:

Launch Anaconda Navigator

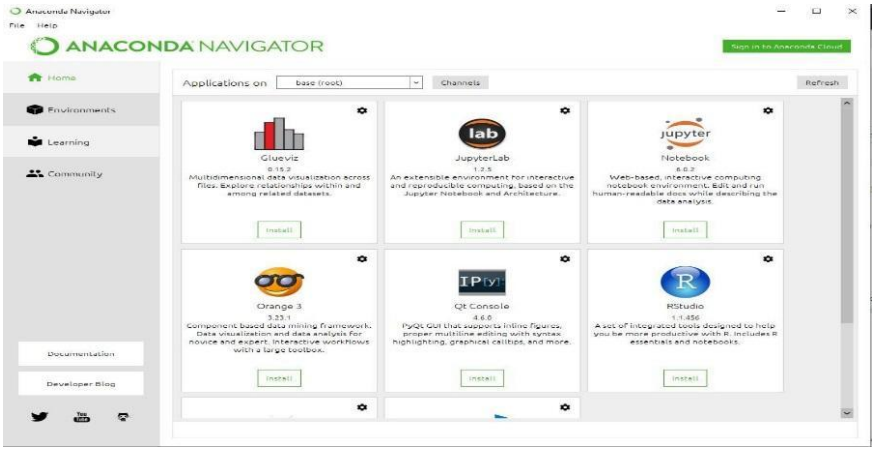

Figure 3.27(a): Anaconda navigator interface (Anaconda, 2020)

(a) **Click on the Install Jupyter Notebook Button:**

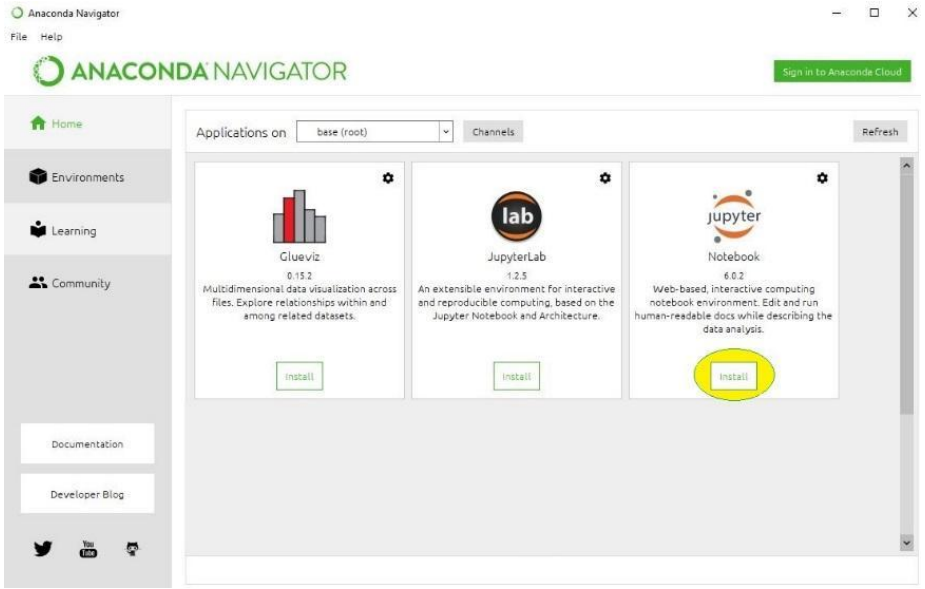

Figure 3.27(b): Anaconda navigator interface (Install) (Anaconda, 2020)

(b) **Beginning the Installation:**

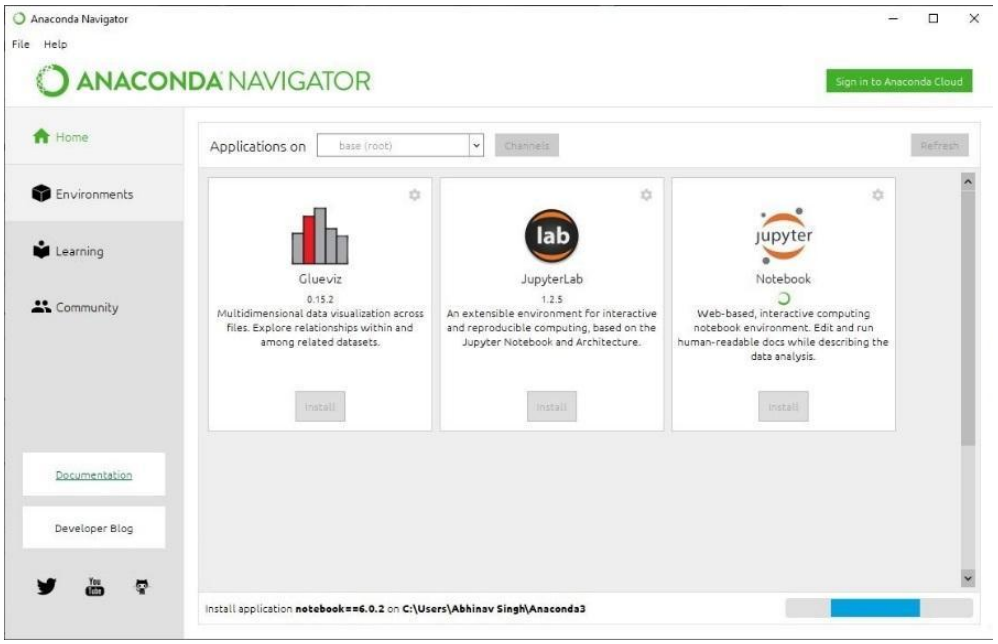

Figure 3.27(c): Anaconda navigator interface (Beginning the Installation) (Anaconda, 2020)

1. **Loading Packages:**

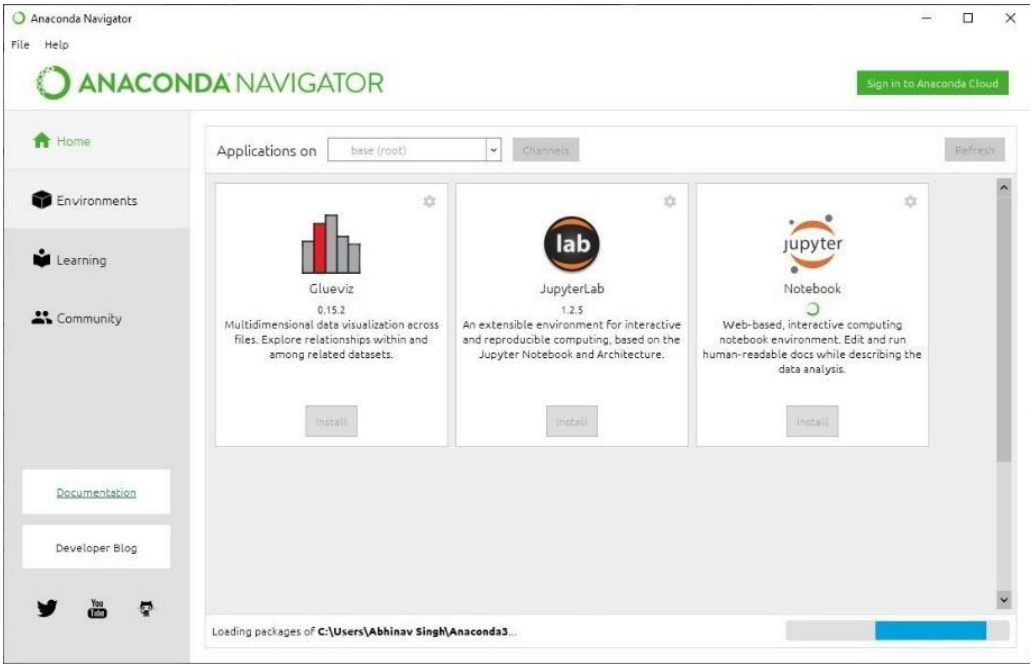

Figure 3.27(d): Anaconda navigator interface (Loading Packages) (Anaconda, 2020)

(c) **Finished Installation:**

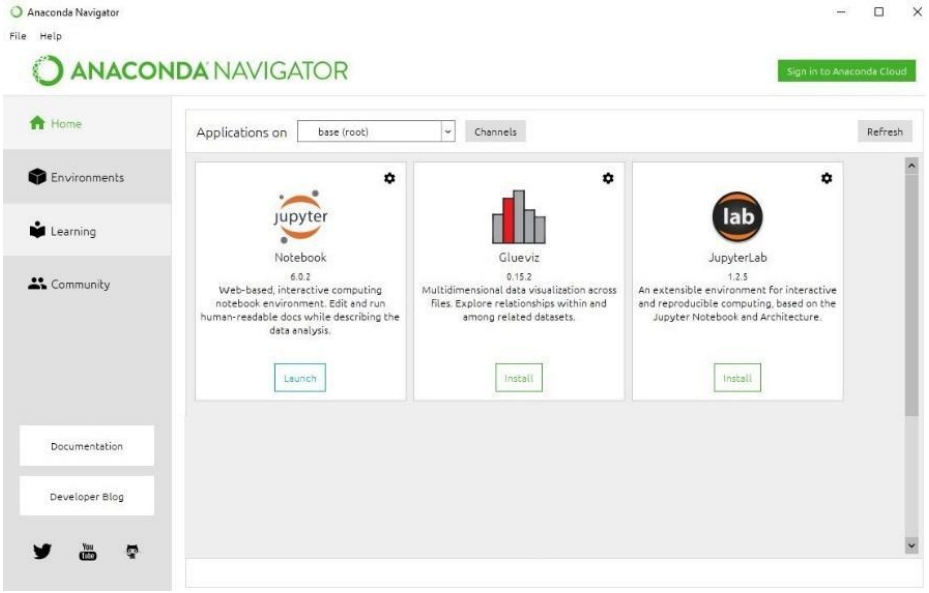

Figure 3.27(e): Anaconda navigator interface (Finished Installation) (Anaconda, 2020)

(d) **Launching Jupyter:**

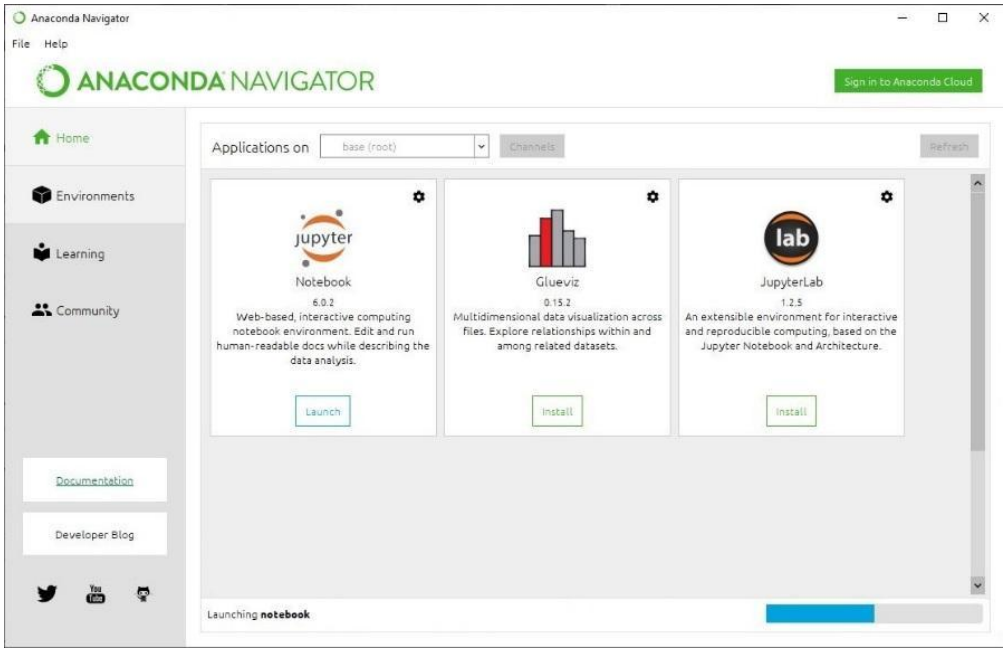

Figure 3.27(f): Anaconda navigator interface (Launching Jupyter) (Anaconda, 2020)

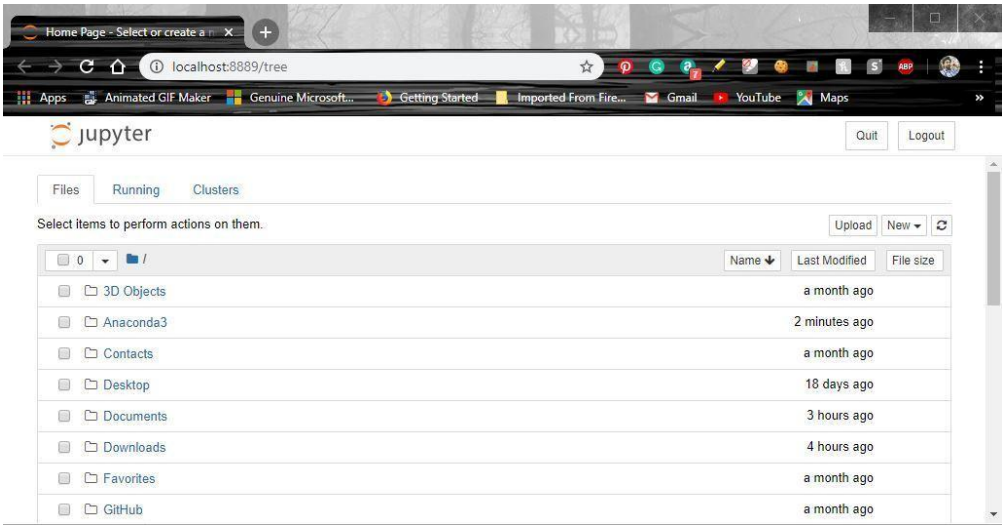

Figure 3.27(g): Jupyter working environment (Jupyter, 2020)

CHAPTER FOUR
RESULTS AND DISCUSSION

This chapter focuses on the various results obtained from the implementation of the proposed application developed. The various results and discussions were focused on the following sections: system interfaces, system testing, system evaluation and system acceptability.

4.1 System interfaces

This section discusses the various interfaces that stemmed out as a result of the final deployment of the developed system. Figures 4.1 – 4.27 depicts the different interfaces developed.

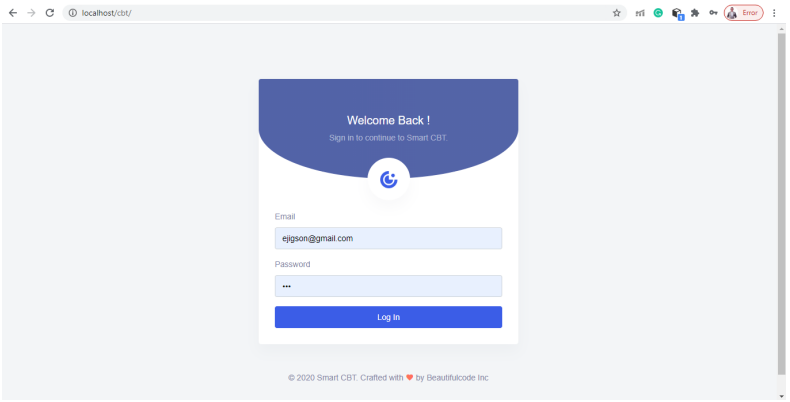

Figure 4.1: Admin login interface
(Researcher, 2020)

Figure 4.1 depicts the admin login web interface. This web page is responsible for authenticating the system administrator

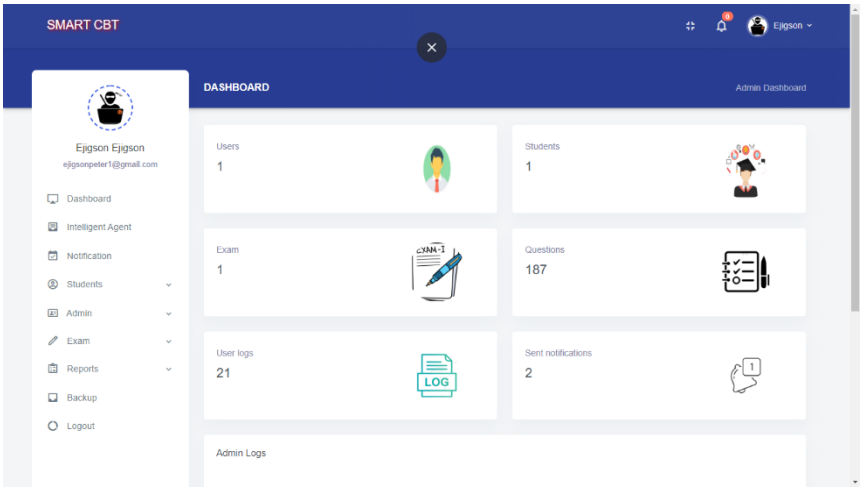

Figure 4.2: Admin dashboard (Researcher, 2020)

Figure 4.2 depicts the admin dashboard web interface. The admin dashboard helps the administrator to have a holistic screen view of some very important activities and events going on in the system like number of users currently logged into the platform, number of students currently using the system for examination, number of questions, number of users log, number of notification sent out, etc. It also contains links and navigation to other functional pages like intelligent agent, notification, student, exam, reports, backup, logout, and so on.

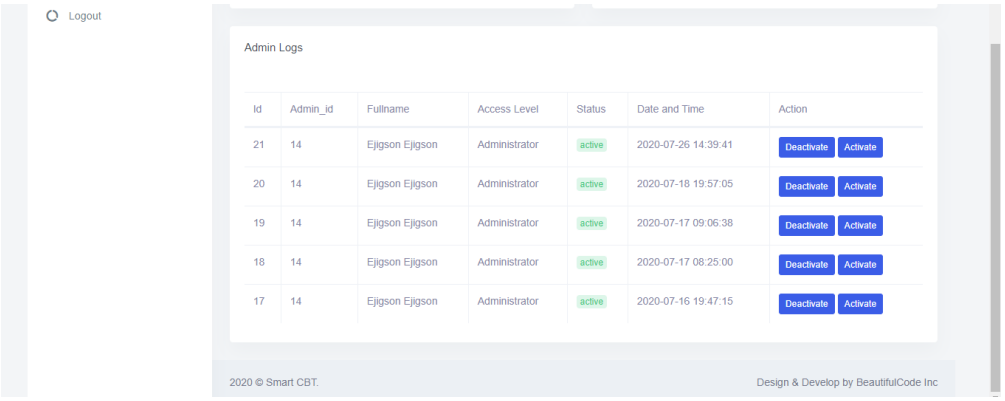

Figure 4.3: Admin log interface (Researcher, 2020)

Figure 4.3 depicts the web interface for the admin log. The log interface helps the administrator to know about the actual persons that have logged into the platform with the following information about the person displayed: full name, access level given, status, date and time they logged into the platform, and finally the action taken.

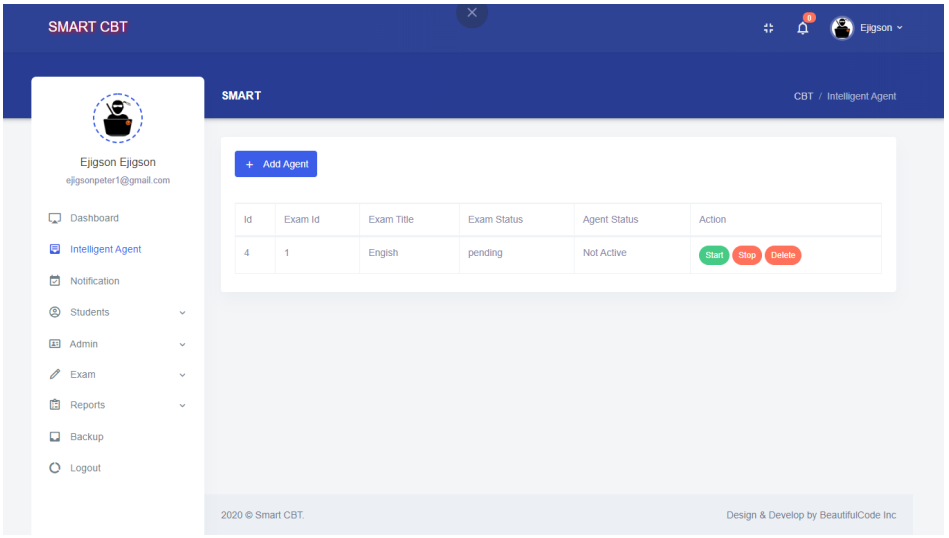

Figure 4.4: Intelligent Agent Interface (Researcher, 2020)

Figure 4.4 shows the intelligent agent interface which is responsible for assigning specific examination roles like starting, stopping or deleting a particular examination schedule.

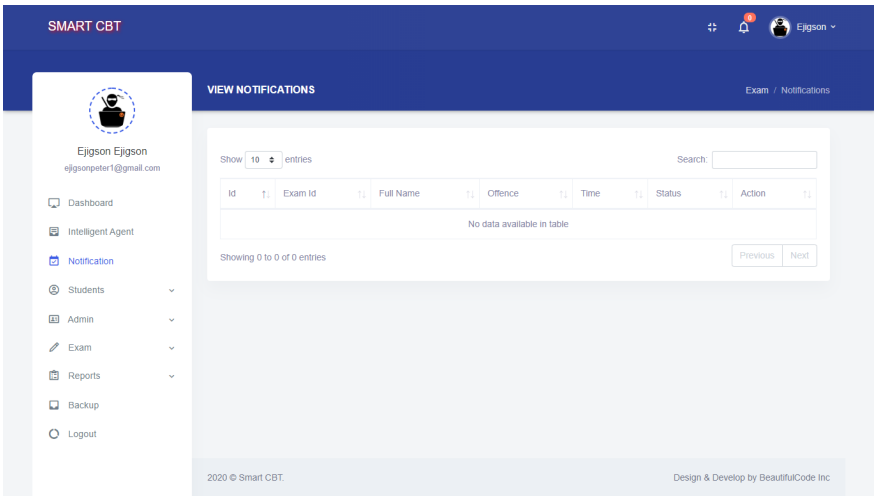

Figure 4.5: Interface for viewing notifications (Researcher, 2020)

Figure 4.5 shows the interface for viewing notifications sent by the intelligent agent to the administrator of list of student offenders during examinations.

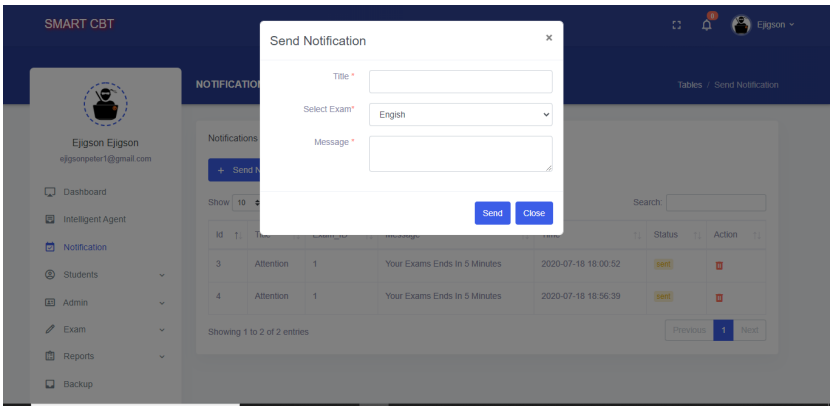

Figure 4.6: Notification interface (Researcher, 2020)

Figure 4.6 shows the notification interface used by the administrator to send messages to students taking a particular test or examination

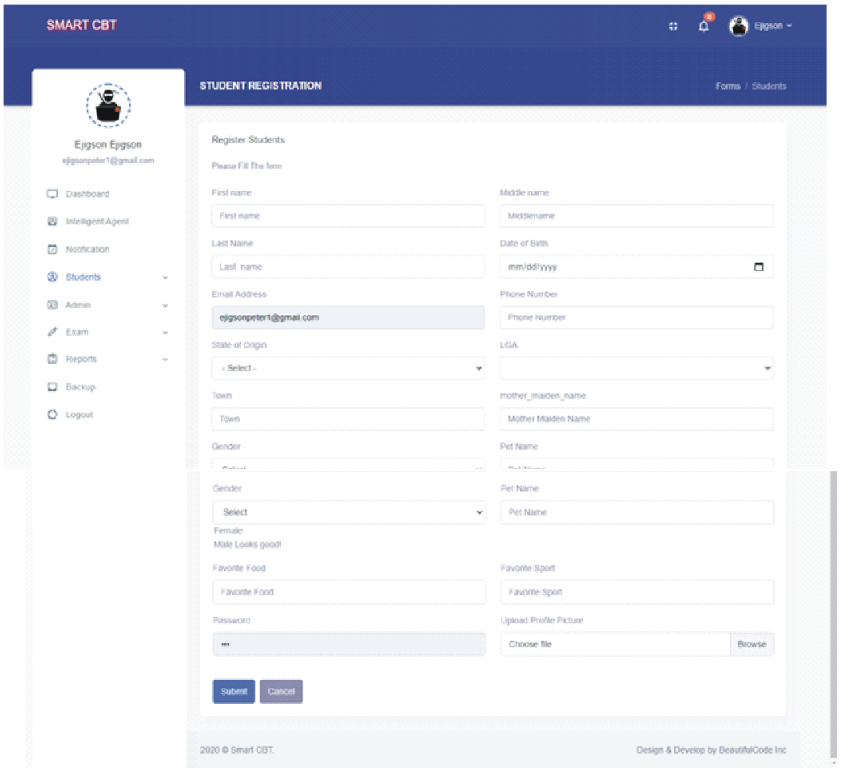

Figure 4.7: Student registration interface (Researcher, 2020)

Figure 4.7 shows the registration interface that allows the administrator to register new students on the platform.

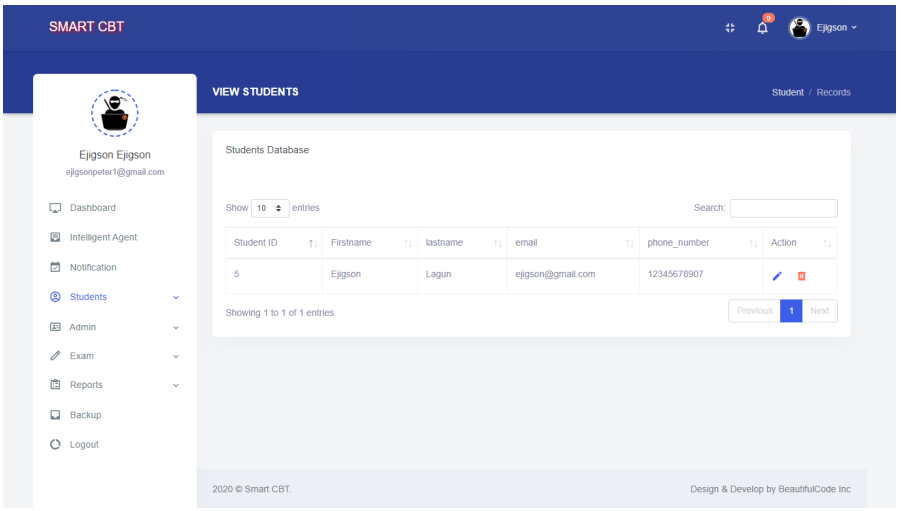

Figure 4.8: view students interface (Researcher, 2020)

Figure 4.8 shows the interface that allows the administrator to view students on the proposed system.

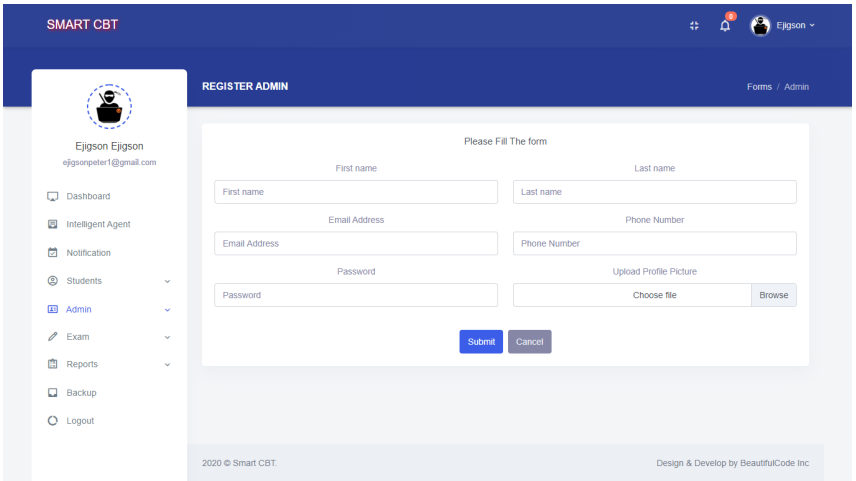

Figure 4.9: Add admin interface (Researcher, 2020)

Figure 4.9 shows the interface that allows the super administrator to create and add administrative roles to staff on the proposed system.

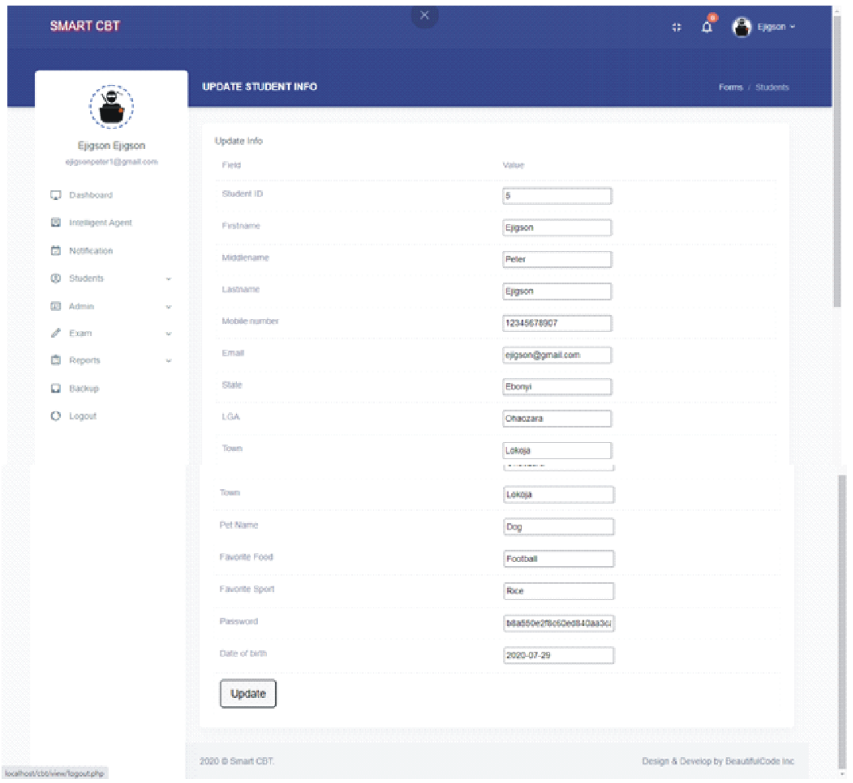

Figure 4.10: Update student record interface (Researcher, 2020)

Figure 4.10 shows the interface that allows the administrator to update student's records or information on the proposed system.

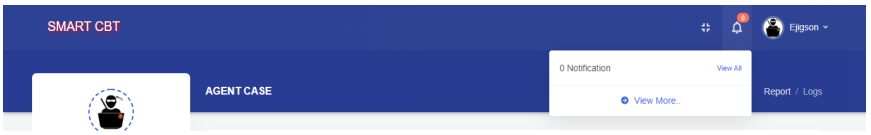

Figure 4.11: Agent notification interface (Researcher, 2020)

Figure 4.11 shows all the notifications sent by the intelligent agent to users on the proposed system.

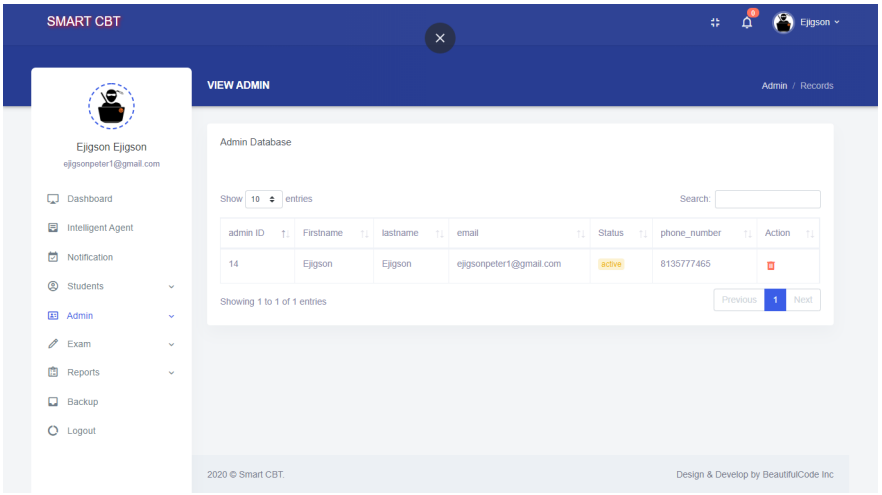

Figure 4.12: View admin interface (Researcher, 2020)

Figure 4.12 shows the interface that allows the administrator to display the records of all administrators on the proposed system.

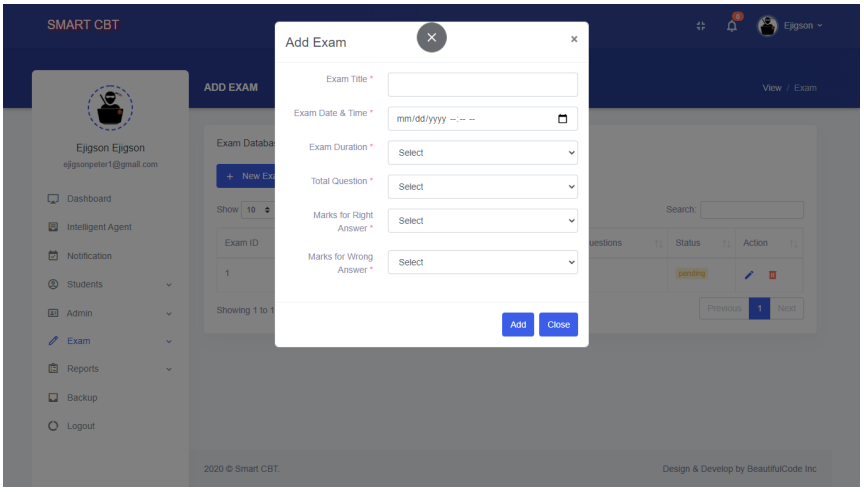

Figure 4.13: Add and view exam interface (Researcher, 2020)

Figure 4.13 shows the interface that allows the administrator to schedule exam and view all exam scheduled on the proposed system.

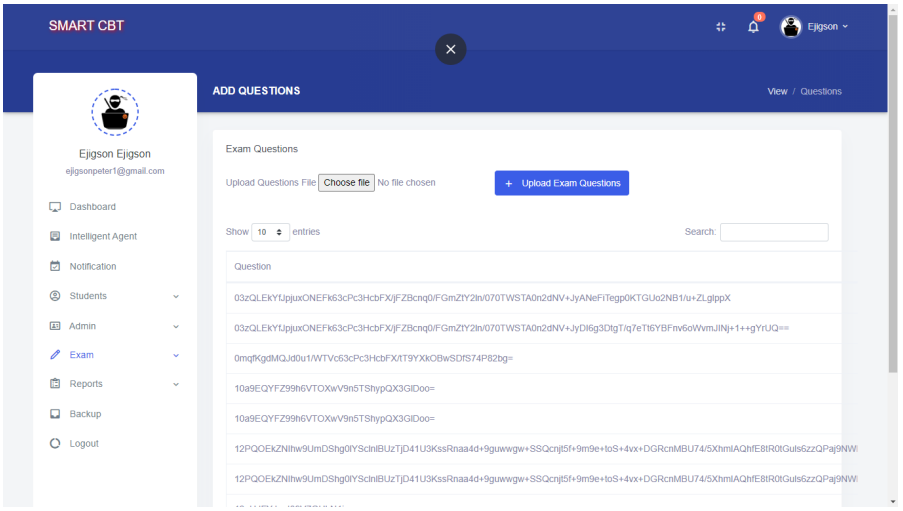

Figure 4.14: Add and view questions interface (Researcher, 2020)

Figure 4.14 shows the interface that allows the administrator to upload and view questions on the proposed system.

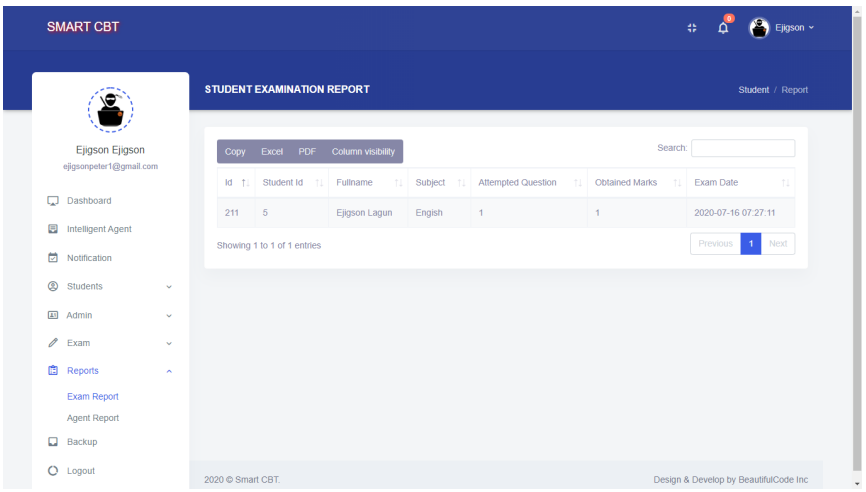

Figure 4.15: Examination report interface (Researcher, 2020)

Figure 4.15 shows the interface that allows the administrator to export examination report to PDF and Excel worksheet files from the proposed system.

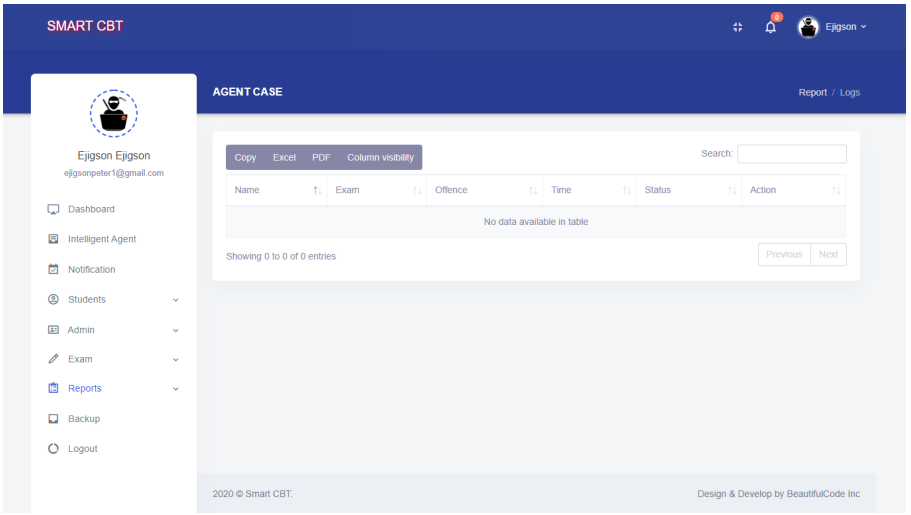

Figure 4.16: Case report interface (Researcher, 2020)

Figure 4.16 shows the interface that allows the administrator to export the data of all reported cases by the intelligent agent on the proposed system.

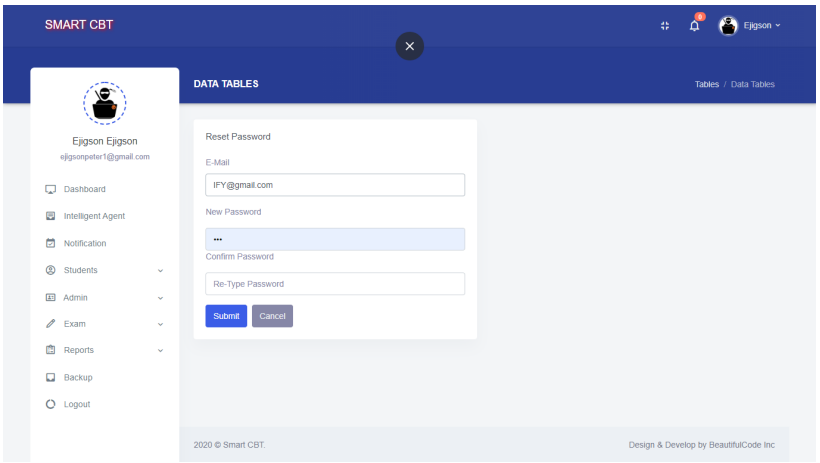

Figure 4.17: Reset password interface (Researcher, 2020)

Figure 4.17 shows the interface that allows the administrator to reset user's password on the proposed system.

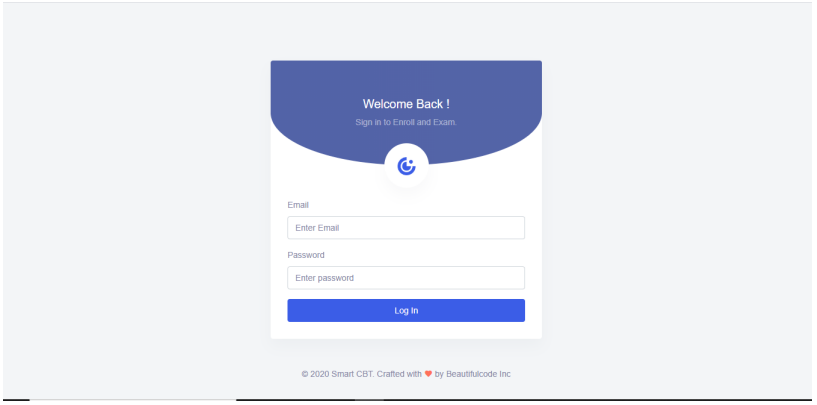

Figure 4.18: Student login interface (Researcher, 2020)

Figure 4.18 shows the interface that allows authentication of student's login credentials the proposed system.

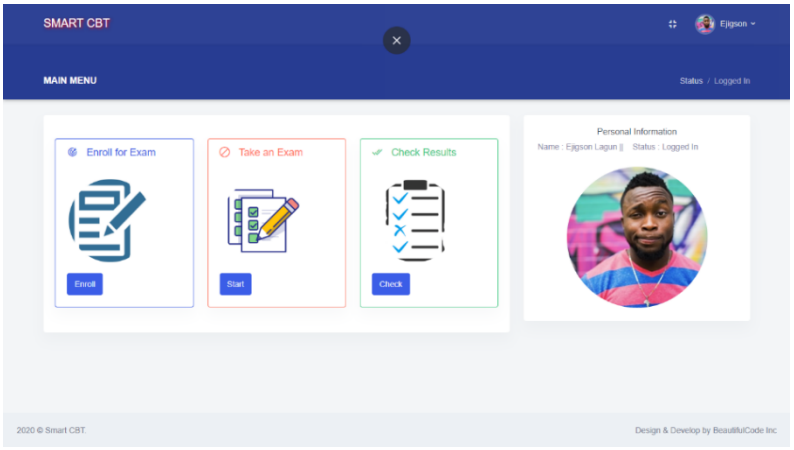

Figure 4.19: Student main menu interface (Researcher, 2020)

Figure 4.19 shows the main menu interface that allows the students to navigate through the student environment on the proposed system.

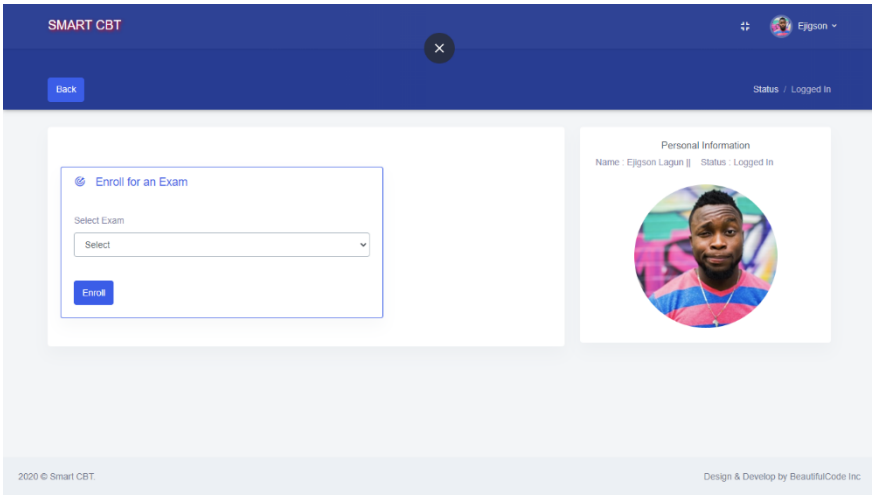

Figure 4.20: Student examination enrollment interface (Researcher, 2020)

Figure 4.20 shows the student's examination enrollment interface that allows the students to enroll for a particular examination on the proposed system.

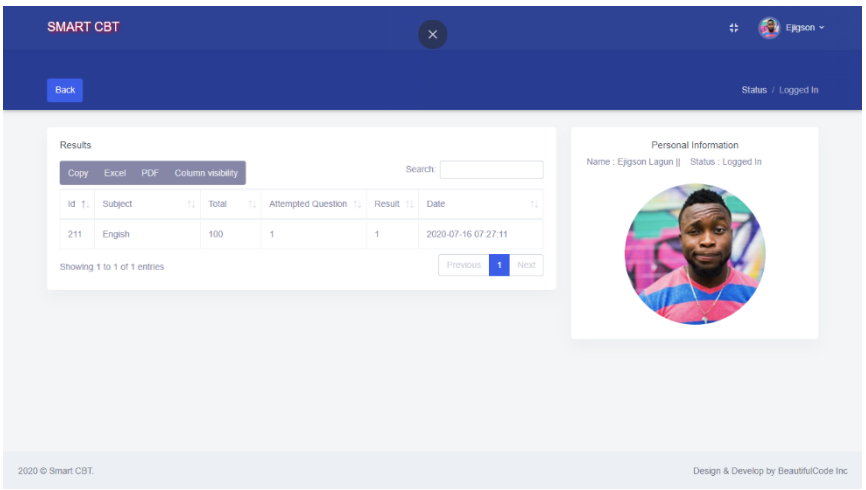

Figure 4.21: check result interface (Researcher, 2020)

Figure 4.21 shows the interface that allows the students to check their results of a particular examination on the proposed system.

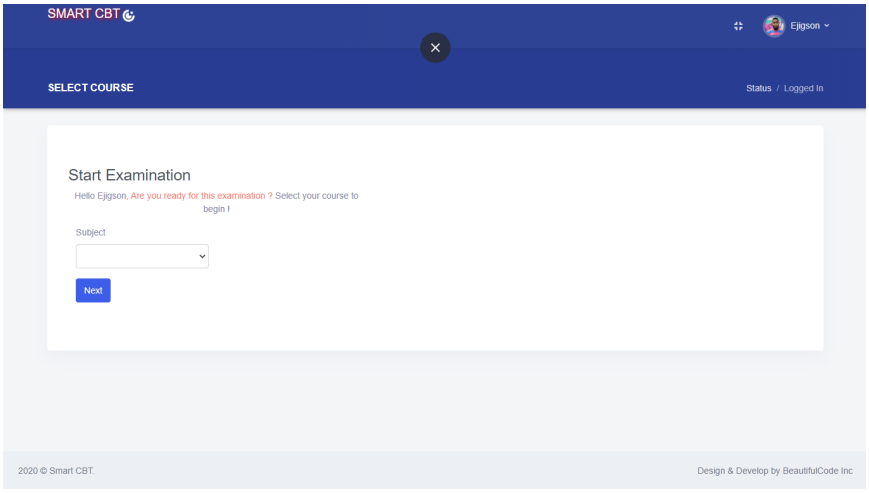

Figure 4.22: Select examination interface (Researcher, 2020)

Figure 4.22 shows the interface that allows a student to select a particular examination from a list of enrollment on the proposed system.

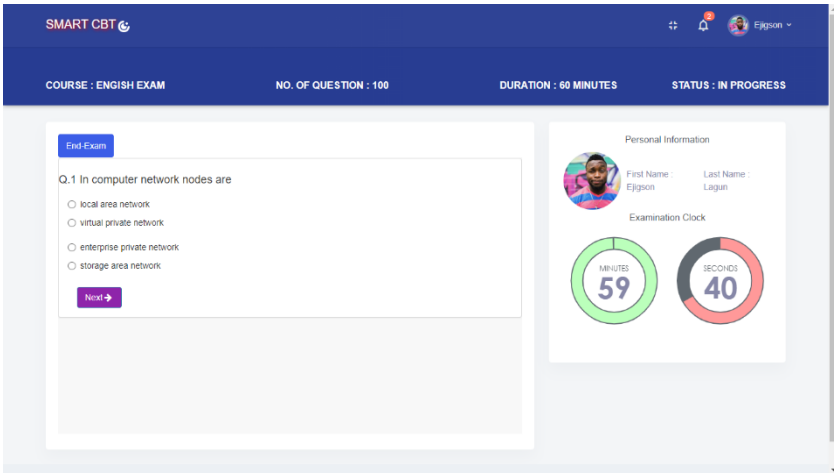

Figure 4.23: Examination interface (Researcher, 2020)

Figure 4.23 shows the interface that allows a student to take an examination on the proposed system.

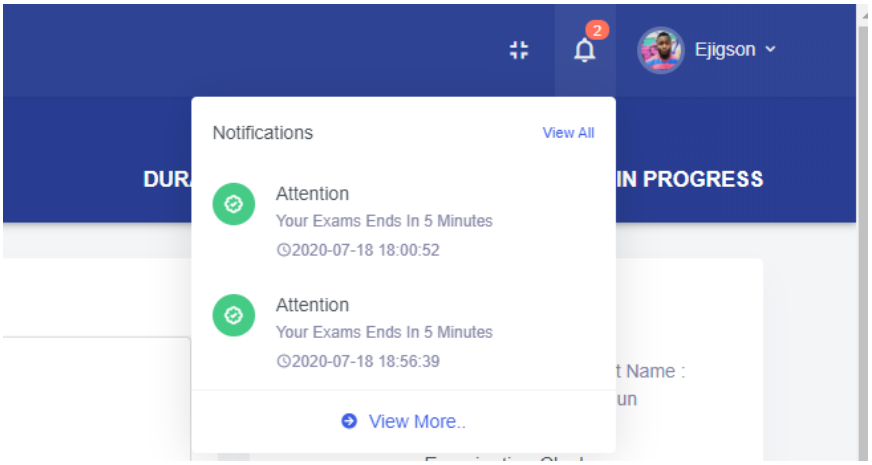

Figure 4.24: Examination notification interface (Researcher, 2020)

Figure 4.24 shows the examination notification interface that displays all notifications sent by the administrator on the proposed system.

Figure 4.25: Examination result interface (Researcher, 2020)

Figure 4.25 shows the examination result interface that allows a student to view his or her results after the examination on the proposed system.

Figure 4.26: Intelligent agent loading interface (Researcher, 2020)

Figure 4.26 shows the intelligent agent loading screen view on the proposed system.

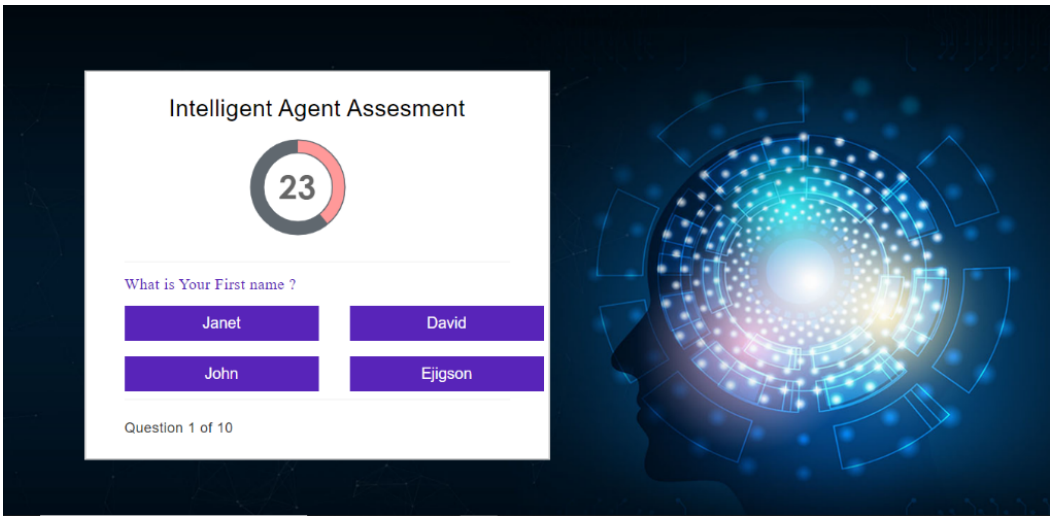

Figure 4.27: Intelligent agent loading interface (Researcher, 2020)

Figure 4.27 shows the intelligent agent assessment interface which is compulsory for students on the proposed system.

4.2 System Testing

This section discusses the different test cases carried out under the developed proposed system in order to ensure that the model is effective. In each test case; the test ID, description, expected result and actual result are provided.

Table 4.1: Admin login test case

Test ID	Description	Expected result	Actual result
1.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button	▪ Expected to grant the admin user access to the admin dashboard	▪ Takes the admin user to the admin dashboard on the proposed model if the admin input entries (email address and password) is correct Else pops up the message "*Invalid login details*".

(Researcher, 2020)

Table 4.2: Student login test case

Test ID	Description	Expected result	Actual result
2.0	1. A student user lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button	▪ Expected to grant the student user access to the student's homepage or dashboard	▪ Takes the student user to the student's homepage or dashboard on the proposed model if the student input entries (email address and password) is correct Else pops up the message "*Invalid login details*".

(Researcher, 2020)

Table 4.3: Activating an examination test case

Test ID	Description	Expected result	Actual result
3.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button. 5. Admin dashboard displays 6. Admin clicks on the intelligent agent link on the dashboard 7. Admin clicks on the Add agent for a particular exam 8. Select the Exam ID 9. Clicks on the agent's start button	▪ Students to write that particular activated examination are expected to see the exam course code on their dashboard in order to commence their online examination.	▪ Students are able to take that particular examination.

(Researcher, 2020)

Table 4.4: Viewing an examination notification test case

Test ID	Description	Expected result	Actual result
4.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button. 5. Admin dashboard displays 6. Admin clicks on the notification link on the dashboard.	▪ It is expected that the following details (ID, Exam ID, Full names, Offence, Time, Status, and Action taken) will be displayed on the admin's screen.	▪ Admin is able to see the ID, Exam ID, Full names, Offences, Time, Status, and Action taken details from the admin dashboard.

(Researcher, 2020)

Table 4.5: Student registration test case

Test ID	Description	Expected result	Actual result
5.0	1. Admin lunches the Smart CBT application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button. 5. Admin dashboard displays 6. Admin clicks on the student link on the dashboard. 7. Admin register student using the pop up registration form. 8. Admin fill in the students details 9. Admin clicks in the submit button	▪ It is expected that the admin can gain access to the Smart CBT platform in order to register students.	▪ Admin is able to gain access to Smart CBT model in order to register students.

(Researcher, 2020)

Table 4.6: Update students records test case

Test ID	Description	Expected result	Actual result
6.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button. 5. Admin dashboard displays 6. Admin clicks on the student's link on the dashboard. 7. Select a particular student to effect change of record/information 8. Admin effect changes 9. Clicks on the UPDATE button.	▪ It is expected that the admin can gain access to the proposed model in order to update students' records.	▪ Admin is able to gain access to proposed model in order to update students.

(Researcher, 2020)

Table 4.7: Reset user's password test case

Test ID	Description	Expected result	Actual result
7.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button. 5. Admin dashboard displays 6. Clicks on the admin link on the dashboard. 7. Clicks on the Reset password	▪ Expected to see a password reset form to input details like user email address, new password, and confirm password entered.	▪ Admin is able to carry out the function of password reset.

(Researcher, 2020)

Table 4.8: Student take examination and view result test case

Test ID	Description	Expected result	Actual result
8.0	1. A student user lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button 5. Student clicks on Enrol Exam to select a particular exam of interest. 6. Student clicks on Take an Exam 7. Student clicks on finish exam 8. Student clicks on view exam results	▪ Student is expected to be granted access to the student's main menu or dashboard in order to take and view the examination taken.	▪ Student is able to take and view examination taken.

(Researcher, 2020)

Table 4.9: Event log test case

Test ID	Description	Expected result	Actual result
9.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button b. Admin dashboard displays 10. Admin clicks on the event log link on the dashboard.	▪ Admin is expected to be granted access to the admin's dashboard in order to view the event log recorded by the proposed application.	▪ Admin is able to view the event log recorded by the proposed application.

(Researcher, 2020)

Table 4.10: Intelligent agent assessment/classification of impersonators test case

Test ID	Description	Expected result	Actual result
10.0	1. Admin lunches the proposed application using any web browser 2. Enter email address 3. Enter password 4. Click on LOGIN button 5. Admin dashboard displays 6. Admin activates the Intelligent agent	▪ It is expected that the intelligent agent lunches and also pops up the agent's assessment interface made up ten questions to be answered by the students one after the other. ▪ Then, the agent will give control to the actual examination to be taken by the student.	▪ A student user is able to take the first assessment by the intelligent agent before starting the real examination based on if the intelligent agent's profiling and classification report is negative. ▪ If the report is positive the agents logs all anomalies and immediately log out the student from attempting the examination reasons being a potential risk of an impersonator.

(Researcher, 2020)

4.3 System Evaluation

This section discusses the proposed system performance evaluation in terms of its ability to detect and properly classify impersonators using model for taking online test or examination and also a comparative evaluation between the developed proposed model and other designed existing model in terms of their features and functionalities.

4.3.1 Developed Proposed System Classification Algorithm Evaluation

The K-Nearest Neighbor machine learning classification algorithm was implemented using Python programming language to detect and properly classify impersonators in our proposed.

A total of 3,083 dataset was gathered, and 80% (2,466 records) of the dataset was used to train the proposed model (See Appendix A - Table A.1), while 20% (617 records) was used in testing the model developed to prevent the problem of over fitting (See Appendix B - Table B.1). Assessing the efficiency of an algorithm embroils using distinct metrics. The confusion metrics, precision, recall and f1 score are the utmost frequently used metrics. In computing the different metrics, the confusion matrix and classification report sklearn.metric methods was implemented using Python programming language.

```
In [31]: from sklearn.metrics import classification_report, confusion_matrix
         print(confusion_matrix(y_test, y_pred))
         print(classification_report(y_test, y_pred))
```

```
[[127   0   1]
 [  0  96   0]
 [  0   0 393]]
              precision    recall  f1-score   support

medium_threat       1.00      0.99      1.00       128
    no_threat       1.00      1.00      1.00        96
       threat       1.00      1.00      1.00       393

     accuracy                           1.00       617
    macro avg       1.00      1.00      1.00       617
 weighted avg       1.00      1.00      1.00       617
```

Figure 4.28: Python report of confusion metrics (Researcher, 2020)

Figure 4.28 revealed the results of the implemented KNN algorithm. Findings also revealed that the KNN algorithm was able to classify all the 617 records in the test dataset with 100% accuracy, which is excellent.

Comparing Error Rate with the K-Value

In comparing the error rate, five (5) values were randomly chosen for the K value and the results gave 100% accuracy. It is observed that one good technique to aid finding the best value of K, is to plot a K value graph and the equivalent error rate for the dataset.

In this section, the mean error and the predicted values of the test dataset for all the K values between 1 and 40 was plotted. In carrying out this task, we had to first calculate the mean of error for all the predicted values where K ranges from 1 and 40 using Python programming language.

```
In [23]: error = []

         # Calculating error for K values between 1 and 40
         for i in range(1, 40):
             knn = KNeighborsClassifier(n_neighbors=i)
             knn.fit(X_train, y_train)
             pred_i = knn.predict(X_test)
             error.append(np.mean(pred_i != y_test))

In [24]: plt.figure(figsize=(12, 6))
         plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
                  markerfacecolor='blue', markersize=10)
         plt.title('Error Rate K Value')
         plt.xlabel('K Value')
         plt.ylabel('Mean Error')

Out[24]: Text(0, 0.5, 'Mean Error')
```

Figure 4.29: Python code segment computing the mean error for k value (Researcher, 2020)

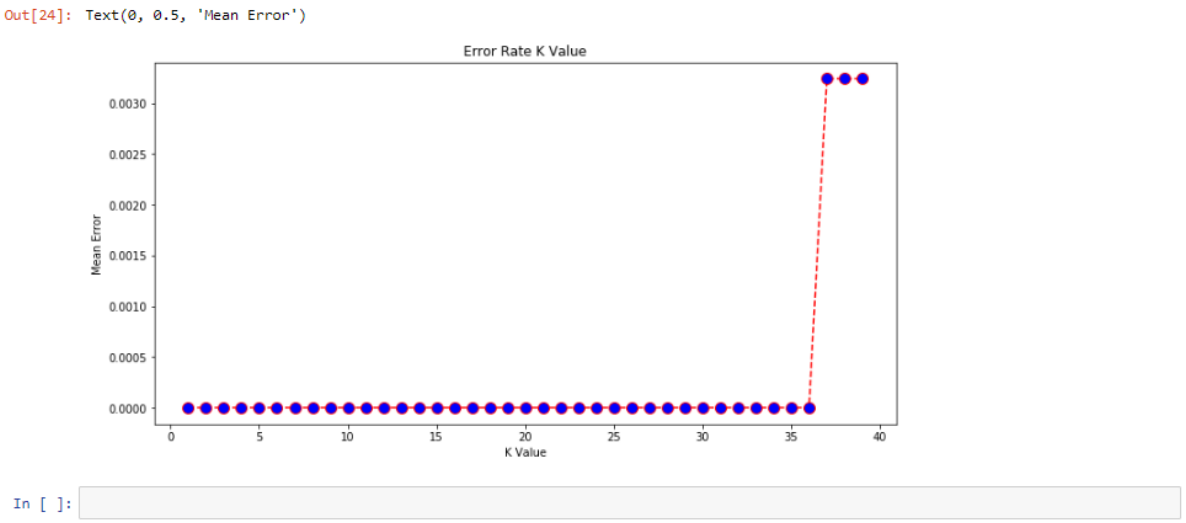

Figure 4.30: Python generated graphical report for the error rate k value (Researcher, 2020)

The Confusion Matrix

A confusion matrix is a swift of predicted results on a particular classification problem. The number of accurate and unfitting predictions are briefed with count values fragmented by each class. The confusion matrix actually pictures the way in which a classification model is disorderly arranged when the model makes a prediction. It provides an understanding to the errors being made and also the types of errors that are being made by a classifier. The error types are as follows:

(a) Class 1 error, which is positive

(b) Class 2 error, which is negative

The confusion matrix terms are denoted as follows:

(i) Positive (P), means there is an affirmative or positive observation

(ii) Negative (N): means there is no affirmative or undesirable observation

(iii) True Positive (TP): means there is affirmative observation, and it is predicted to be affirmative.

(iv) False Negative (FN): means there is affirmative observation, but it is predicted to be undesirable or negative.

(v) True Negative (TN): means there is undesirable or negative observation, and it is predicted to be undesirable.

(vi) False Positive (FP): means there is undesirable or negative observation, but it is predicted affirmative or positive.

Hence, the Confusion Matrix is generated using Python programming language as follows

[124 0 0 0 77 0 0 0 416]

Detailed breakdown of the Confusion Matrix:

Table 4.11(a): Confusion Matrix Predictions

N = 617	Predicted: medium_threat	Predicted: no_threat	Predicted: threat
Actual: medium_threat	124	0	0
Actual: no_threat	0	77	0
Actual: threat	0	0	416

(Researcher, 2020)

Table 4.11(b): Confusion Matrix False Rates:

N = 617	Predicted: medium_threat	Predicted: no_threat	Predicted: threat	
Actual: medium_threat	TP = 124	FP = 0	FP = 0	124
Actual: no_threat	FN = 0	TP = 77	FN = 0	77
Actual: Threat	FN = 0	FN = 0	TN = 416	416
	124	77	416	

(Researcher, 2020)

Computation of Classification Rate / Accuracy

Classification Rate or Accuracy is given by the relation:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

(1)

However, there are some of the problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be referred to as been excellent, good, mediocre, poor or terrible depending upon the problem.

$$\text{Accuracy} = \frac{(TP + TP + TN)}{(TP + TP + TN + FP + FN + FN + FN + FN + FN)}$$

(2)

$$\text{Accuracy} = \frac{(124+77+416)}{(124+77+416+0+0+0+0+0+0)} \quad (3)$$

$$\text{Accuracy} = 1.00$$

Computation of Recall

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN). Recall gives us an idea about when it is actually a yes, how often does it predict yes.

$$\text{Recall (No Threat)} = \frac{(TP)}{(TP + FN)}$$

(1)

$$\text{Recall (No Threat)} = \frac{(77)}{(77 + 0)} \quad (2)$$

$$\text{Recall (No Threat)} = 1.00$$

Computation of Precision

To compute the value of precision; we had to divide the total number of correctly classified positive examples by the total number of predicted positive examples. High

Precision indicates an example labelled as positive is indeed positive (a small number of FP).

High recall, low precision:

This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

Low recall, high precision:

This shows that we miss a lot of positive examples (high FN) but those we predicted as being positive are indeed really positive (low FP).

The precision expresses that when it predicts YES, how often is it correct.

$$\text{Precision (No Threat)} = \frac{(TP)}{(TP + FP)} \quad (1)$$

$$\text{Precision (No Threat)} = \frac{(77)}{(77 + 0)} \quad (2)$$

$$\text{Precision (No Threat)} = 1.00$$

Computation of F-measure:

Since we have two measures i.e. (Precision and Recall) it helps to have a measurement that represents both of them. In calculating the F-measure (F1-Score) which uses the Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

It is observed that the F-Measure (F1-Score) will always be closer to the lesser value of the Precision or Recall values.

$$\text{F1-score (F-Measure)} = \frac{(2 * recall * precision)}{(recall + precision)} \quad (1)$$

$$\text{F1-score (F-Measure)} = \frac{(2 * 1.0 * 1.0)}{(1.0 + 1.0)} \quad (2)$$

$$\text{F1-score (F-Measure)} = 1.00$$

The overall results revealed that the KNN algorithm used in the implementation of impersonators detection and classification mechanism of the proposed model was able to classify all the 30 records in the test dataset with 100% accuracy, which is excellent.

4.3.2 System Comparative Evaluation

Table 4.12: Features/functionalities comparative evaluation

S/N	Authors / CBT Models	Features / Functionalities											
		Username & Password Login	Secured authentication	Triple-A Model Method	Secured Database	Randomized Questions	Multiple Choice Questions (MCQ)	Agent Message Notification System	Different Report Generation Formats	Bulk Upload of Questions	Secured code for examinable courses	Event logs	Detection and Classification
1.	Fagbola, Adigun and Oke (2013)	√	X	√	X	√	√	X	√	√	X	√	X
2.	Ajinaja (2017)	√	X	√	X	√	√	X	√	√	X	√	X
3.	Suleiman and Nachandiya (2018)	√	X	√	√	√	√	X	√	√	X	√	X
4.	Omotehinwa and Durojaye (2013)	√	√	√	√	√	√	X	√	X	X	√	X
5.	Ismail and Soye (2018)	√	√	√	√	√	√	X	√	√	X	√	X
6.	Al-Saleem and Ullah (2014)	√	√	√	X	√	√	X	√	X	X	√	X
7.	Adebayo and Abdulhamid (2009)	√	X	√	√	√	√	X	√	X	X	√	X
8.	Wang et al. (2004)	√	X	√	X	√	√	X	√	X	X	√	X
9.	Kuyoro et al. (2016)	√	X	√	X	√	√	X	√	√	X	√	X
10.	Oluwule (2015)	√	X	√	X	√	√	X	√	√	X	√	X
11.	Propose Smart CBT Model	√	√	√	√	√	√	√	√	√	√	√	√

(Researcher, 2020)

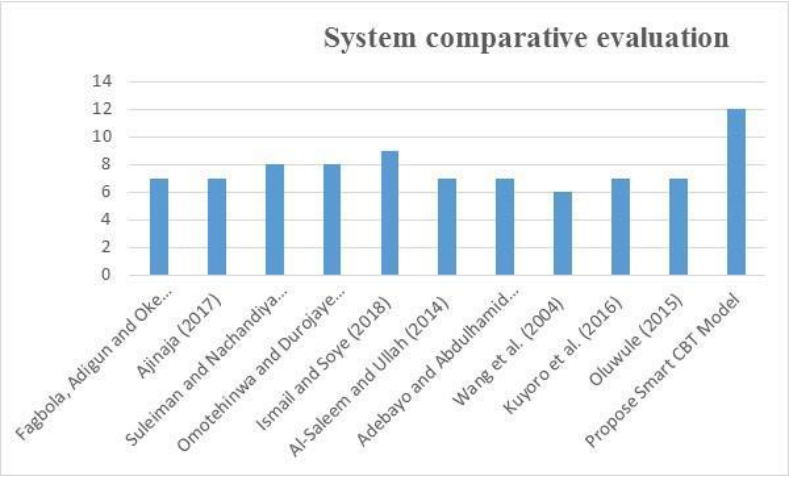

Figure 4.31: System comparative evaluation (Researcher, 2020)

Figure 4.31 shows the rated functionalities of ten (10) different online examination models and the proposed system in a bar chart. Findings further revealed that the proposed model has more functionality as compared to others.

4.4 System Acceptability

This section discusses how the system acceptability test was carried out on the developed proposed application. A structured questionnaire comprising of ten (10) direct questions like ease of usage, user friendly interface design, efficient user's experience design, accessibility, effective security measures integration, effective report generation module, speed of information retrieval, effectiveness of the impersonator detection mechanism, accuracy of grade computation, and finally cost effectiveness on a five (5) point scale response rating system such as Strongly agree, Agree, Indifference, Disagree, Strongly disagree administered to 60 users in order to ascertain the usability and system acceptability.

Sixty (60) responses were collected, of which 55 were valid and 5 were invalid due to users' selection of multiple entries or responses which is unacceptable for the analysis and evaluation.

Table 4.13 revealed the summary of user's responses from the structured questionnaire, while Figures 4.32 – 4.41 revealed the graphical interpretations of the user's responses.

Table 4.13: Summary of users' responses from the questionnaire

S/N	Criteria	Strongly Agree	Agree	Indifference	Disagree	Strongly Disagree
1.	Ease of usage	45	05	05		
2.	User friendly interface design	50	03	02		
3.	Efficient user's experience design	52	03			
4.	Accessibility	48	02	05		
5.	Effective security measures integration	52	02	01		
6.	Effective report generation module	40	08	05	02	
7.	Speed of information retrieval	45	05	05		
8.	Effectiveness of the impersonator detection mechanism	50	05			
9.	Accuracy of grade computation	45	10			
10.	Cost effectiveness	48	02	05		

(Researcher, 2020)

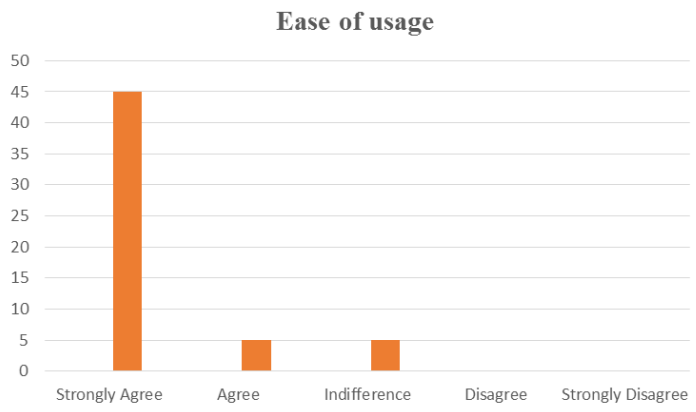

Figure 4.32: Ease of usage analysis (Researcher, 2020)

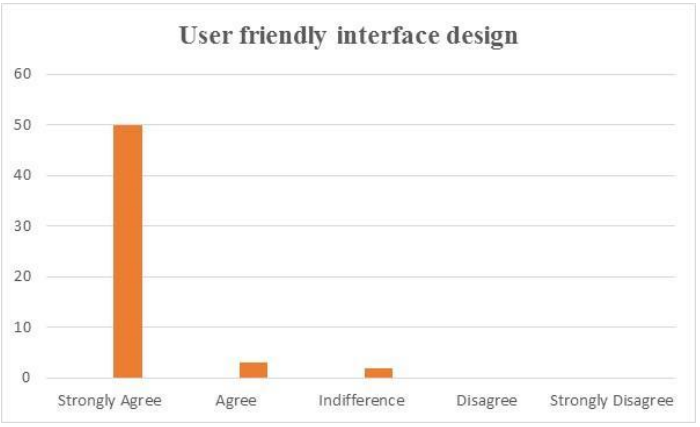

Figure 4.33: User friendly interface design analysis (Researcher, 2020)

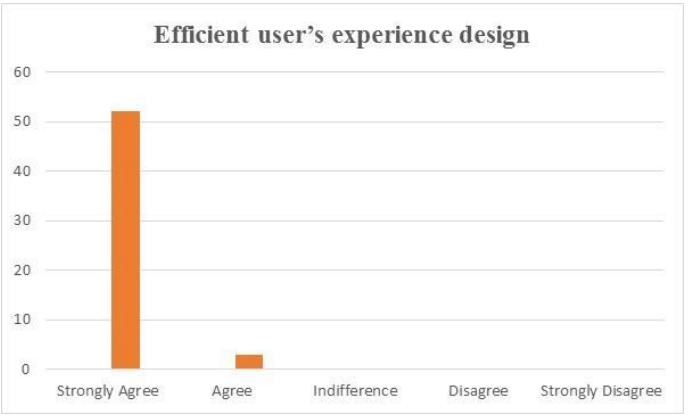

Figure 4.34: Efficient user's experience design analysis (Researcher, 2020)

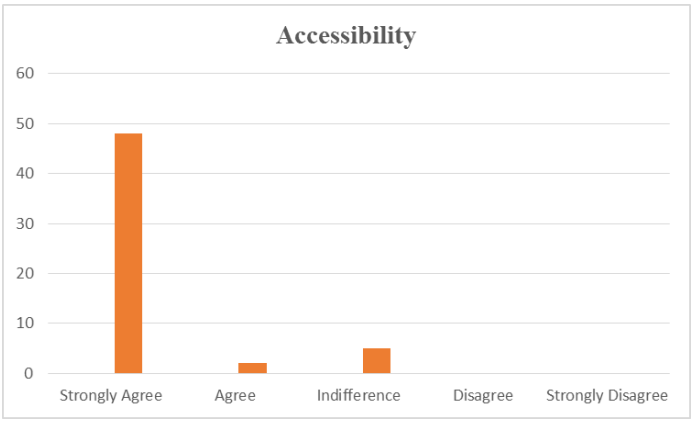

Figure 4.35: Accessibility analysis (Researcher, 2020)

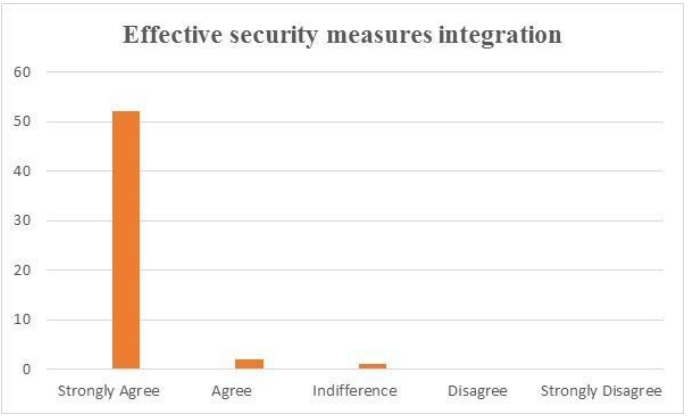

Figure 4.36: Effective security measures integration analysis (Researcher, 2020)

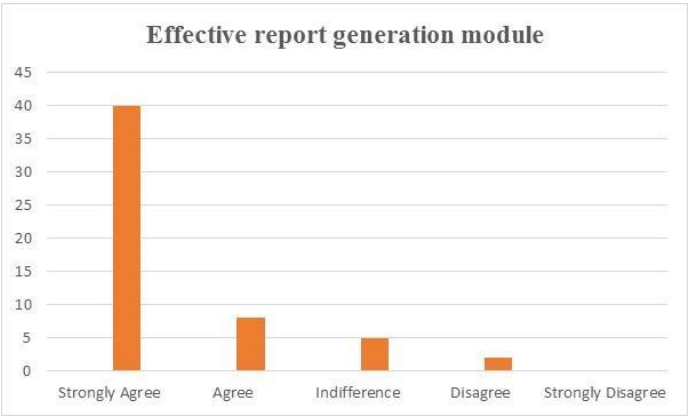

Figure 4.37: Effective report generation analysis (Researcher, 2020)

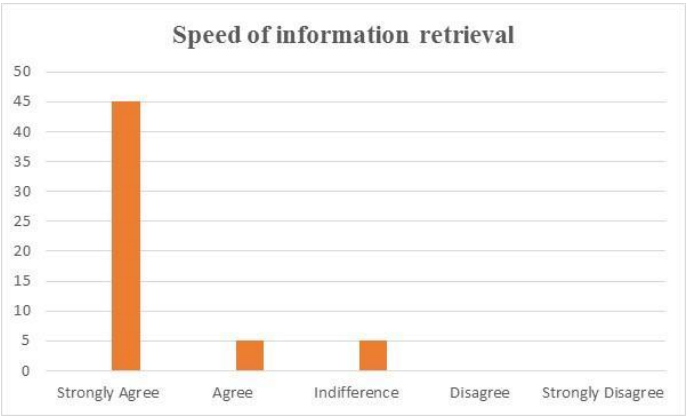

Figure 4.38: Speed of information retrieval analysis (Researcher, 2020)

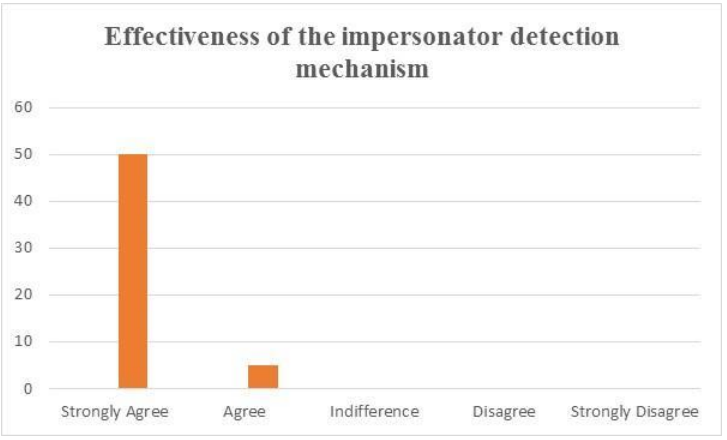

Figure 4.39: Effectiveness of the impersonator detection mechanism analysis

(Researcher, 2020)

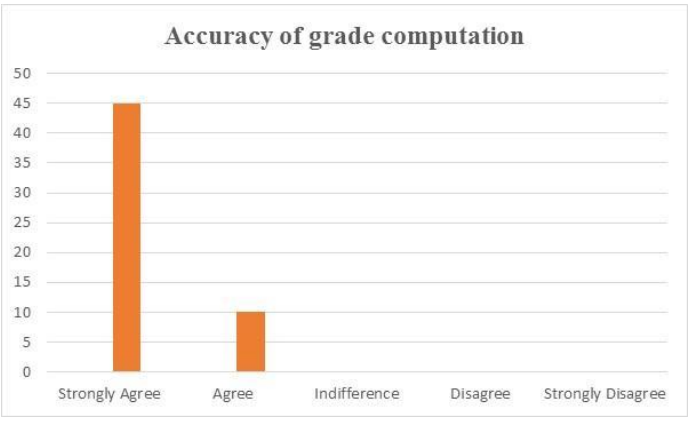

Figure 4.40: Accuracy of grade computation analysis (Researcher, 2020)

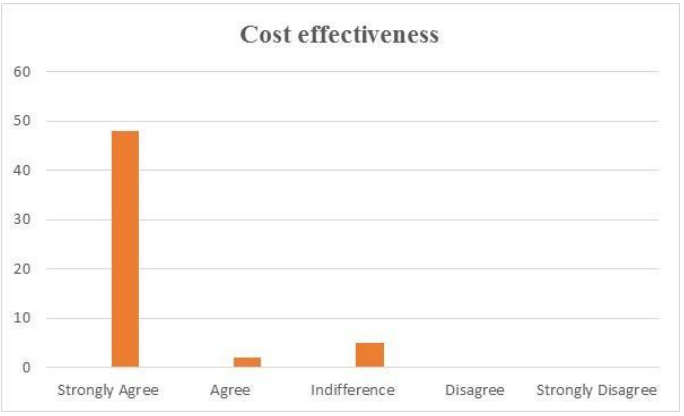

Figure 4.41: Cost of effectiveness analysis (Researcher, 2020)

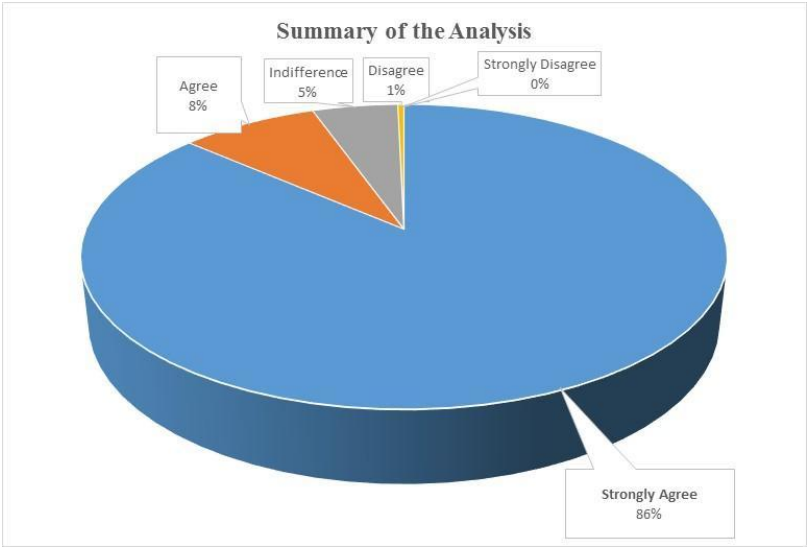

Figure 4.42: Summary of user's responses analysis (Researcher, 2020)

Figure 4.42 depicts a pie chart representing a summary of the entire users' responses in percentage on the 5 point rating scale. 86% of the 55 valid users strongly agrees with the developed proposed model, 8% of the users just agrees with the model, 5% are indifferent about the model, 1% disagrees with the model, while 0% strongly disagrees with the proposed model. This acceptability test revealed that the users were satisfied with the model developed model based on its effectiveness.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The challenges of computer based testing platforms cannot be over emphasized. The existing online examination platforms observed had serious security issues most especially in the areas of login access, database security, and detection of impersonators. In order to reduce or totally eradicate the security issues with the existing system a prototype platform was designed and developed by integrating an intelligent agent services to the conventional online examination model. The proposed system could conveniently process the following: secure logins using 256-bit SSH algorithm, randomly generate questions using the Merssene Twister Algorithm, secure examination questions stored in the database using the 128-Bit Advanced Encryption Standard, secure the examination unique codes using the MD5 algorithm, and finally, detect and properly classify the case of an impersonator during the online examination using the K – Nearest Neighbor (KNN) Algorithm with 99% accuracy as compared to the research work done on online examination system by authors such as (Fagbola, Adigun and Oke, 2013; Ajinaja, 2017; Suleiman and Nachandiya, 2018; Ismail and Soye, 2018; Omotehinwa and Awojaye, 2013; Al-Saleem and Uttah, 2014; Adebayo and Abdulhamid, 2009; Wang et al., 2004; Kuyoro et al., 2016; Oluwule, 2015) that does not have these features in their various CBT systems developed.

5.2 Recommendations

The propose platform was developed using the agile software methodology that supports team collaboration for effective product delivery. For effective utilization of the propose Smart CBT platform developed; we recommend the following:

(a) The adoption of the Smart CBT solution for commercial production and utilization.

(b) The adoption of the Smart CBT platform by universities and other higher institution for proctoring online examinations and same time curbing examination malpractices such as impersonation.

(c) Future research work should consider the integration adaptive testing feature, audio/video scenario based questions feature, and the usage of multi-agent system (MAS) that could possibly handle different services in the same CBT platform or environment.

5.3 Contribution to knowledge

This research work on proposed platform for securing and proctoring online examination contributes to knowledge as follows:

(a) The introduction of new multi-level security measures into the CBT platform

(b) The introduction of machine learning technique to the CBT platform, which makes the platform truly smart for effective detection of impersonators during the online examination and the threat severity level classification.

REFERENCES

Abedi, J. (2014). The use of computer technology in designing appropriate test accommodations for English language learners. Applied Measurement in Education, 27(4), 261-272. doi: 10.1080/08957347.2014.944310

Adebayo, O., and Abdulhamid, S. M. (2009), E-Exams System for Nigerian Universities with Emphasis on Security and Results Integrity, International Journal of the Computer, the Internet and Management, 18(2): 47.1 – 47.12.

Adebayo, O., and Abdulhamid, S. M.(2014). E- Exams System for Nigerian Universities with Emphasis on Security and Result Integrity. International Journal of the Computer, the Internet and Management, 18(2), 47.1-47.12. 2014.

Adegbija, M. V., Fakomogbon, M. A., and Daramola, F. O. (2012). The New Technologies and the Conduct of E-Examinations: A Case Study of National Open University of Nigeria. British Journal of Science, 3(1): 59 - 66.

Adesina, A., Stone, R., Batmaz, F., & Jones, I. (2014). Touch Arithmetic: A process-based Computer-Aided Assessment approach for capture of problem solving steps in the context of elementary mathematics. Computers & Education, 78, 333-343. doi:10.1016/j.compedu.2014.06.015

Adewale, O. A., Ajadi, T. O., and Inegbedion, J. O. (). Perception of learners on Electronic Examination in Open and Distance Learning Institutions: A Case Study of National Open University of Nigeria.

Ajinaja, M. (2017), The Design and Implementation of a Computer Based Testing System using Component Based Software Engineering, International Journal of Computer Science and Technology, 8(1): 58 – 65.

Akinsanmi, O., Agbaji, O. T. R., and Soroyewun, M. B. (2010). Development of an E-Assessment Platform for Nigerian Universities, Research Journal Applied Sciences, Engineering and Technology 2(2): 170-175.

Allen, M. J., & Yen, W. M. (1979). Introduction to measurement theory. Monterey, CA: Brooks/Cole.

Al-Saleem, S. M., and Ullah, H. (2014), Security Considerations and Recommendations in Computer Based Testing, The Scientific World Journal, Hindawi Publishing Corporation, 2014: 1 – 7.

Ayo C. K. (2007). The Prospects of e-Examination Implementation in Nigeria. Turkish Online Journal of Distance Education-TOJDE 2007; 8: Article 10.

Ayo, C. K., Akinyemi, I. O. Adebiyi, A. A., and Ekong, U. O. (2007), The Prospects of E-Examination Implementation in Nigeria, Turkish Online Journal of Distance Education, (8)4: 125 – 135.

Brown, H. D., & Abeywickrama, P. (2010). Language assessment: Principles and classroom practices. White Plains, NY: Pearson Education.

Chou, C., Moslehpour, M., and Le Huyen, N. T. (2014). Concurrent and Predictive Validity of Computer-adaptive Freshman English Test for College Freshman English in Taiwan. International Journal of English Language Education, 2(1), pp-143. doi:10.5296/ijele.v2i1.4919

Chua, Y. P. (2012). Effects of computer-based testing on test performance and testing motivation. Computers in Human Behavior, 28, 1580–1586. doi:10.1016/j.chb.2012.03.020

Chua, Y. P., and Don, Z. M. (2013). Effects of computer-based educational achievement test on test performance and test takers' motivation. Computers in Human Behavior, 29(5), 1889-1895. doi:10.1016/j.chb.2013.03.008

Clariana, R. and Wallace, P. (2002). Paper-based versus computer-based assessment: key factors associated with the test mode effect. British Journal of Educational Technology, 33(5), 593–602. doi:10.1111/1467-8535.00294

Cristobal, R, Sebastian, V., and, Paul, D. B. (2009). Using Mobile and Web-Based Computerized Tests to Evaluate University Students, Wiley Periodicals Inc, Computer Application Eng Education 17(1): 435 – 447.

Debuse, J. C., & Lawley, M. (2015). Benefits and drawbacks of computer-based assessment and feedback systems: Student and educator perspectives. British Journal of Educational Technology. doi:10.1111/bjet.12232

Delen, E. (2015). Enhancing a Computer-Based Testing Environment with Optimum Item Response Time, Eurasia Journal of Mathematics, Science & Technology Education, 11(6), 1457-1472.

Fagbola, T. M., Adigun, A. A., and Oke, A. O. (2013), Computer Based Test (CBT) System for University Academic Enterprise Exam, International Journal of Scientific and Technology Research, 2(8): 336 – 342.

Greiff, S., and Martin, R. (2014). Computer-Based Testing, 2014.

Hedberg, C., Davis, B., Lundeen, B., and Vitella, S. (1995). Computer-Based Testing-Why, What and When. OSEAS-Europe, Athens Conference Minutes.

Ismail, H, M., and Soye, B. M. (2018), Biometric Enabled Computer Based Testing System with Advanced Encryption Standard (AES), JETIR, 5(8): 1 – 8.

Ismail, M., and Shami, P. A. (2012). Computer-Based Vs Paper-Based Examinations: Perception of University Teachers, The Turkish Online Journal of Educational Technology, 11(4): 371 – 381

Jeong, H. (2014). A comparative study of scores on computer-based tests and paper-based tests. Behaviour & Information Technology, 33(4), 410-422. doi:10.1080/0144929X.2012.710647

Jimoh, R. G., AbdulJaleel, K. S., and Kawu, Y. (2012). Students' Perception of Computer Based Test (CBT) for Examining Undergraduate Chemistry Courses, Journal of Emerging Trends in Computing and Information Sciences, 3(2): 125 – 134.

Kaya, F. & Delen, E. (2014). A computer-based peer nomination form to identify gifted and talented students. The Australasian Journal of Gifted Education. 23(2), 29-36.

Kuyoro, S. O., Maminor, G. U., Kanu, R. U., and Akande, O. (2016), The Design and Implementation of a Computer-Based Testing System, Journal of Applied Computation, 1(1): 1 – 7.

Lee, Y. H., & Jia, Y. (2014). Using response time to investigate students' test-taking behaviors in a NAEP computer-based study. Large-scale Assessments in Education, 2(1), 1-24. doi:10.1186/s40536-014-0008-1

Newhouse, C. P. (2013). Computer-Based Exams in Schools: Freedom from the limitations of paper?Research and Practice in Technology Enhanced Learning. (8)3, 431－447.

Olea, J., Abad, F. J., Ponsoda, V., Barrada, J. R., & Aguado, D. (2011). eCAT-listening: Design and psychometric properties of a computer-adaptive test on English listening. Psicothema, 23(4), 802-807.

Oluwole, O. (2015), Design and Implementation of an Open-Source Computer-Based Testing System with End User Impact Analysis in Africa, International Journal of Modern Education and Computer Science, 8(1): 17 – 24.

Omotehinwa, T. O., and Durojaye, D. S. (2013), Computer Based Test: Security and Results Integrity, International Journal of Computer and Information Technology, 2(2): 324 – 329.

Sadiq, F. I., and Onianwa, C. U. (2011), Towards a Scalable Web Assessment System for Post University Matriculation Examination in Nigeria. African Journal of Computer & ICTs Vol. 4, No. 2. pp 25-30. Retrieved on October 4th 2013 from http://www.ajocict.net.

Schatz, P., & Browndyke, J. (2002). Applications of computer-based neuropsychological assessment. Journal of Head Trauma Rehabilitation, 17(5), 395-410.

Schatz, P., & Putz, B. O. (2006). Cross-validation of measures used for computer-based assessment of concussion. Applied Neuropsychology, 13(3), 151-159. doi:10.1207/s15324826an1303_2

Suleiman, A., and Nachandiya, N. (2018), Computer Based Testing (CBT) System for GST Exams in Adamawa State University, Mubi, Asian Journal of Research in Computer Science, 2(1): 1 – 11.

Taşci, T., Parlak, Z., Kibar, A., Taşbaşi, N., and Cebeci, H. I. (2014). A Novel Agent-Supported Academic Online Examination System. Educational Technology & Society, 17 (1), 154–168.

Taylor, A. R. (2005). A Future in the Process of Arrival: Using Computer Technologies for the Assessment of Student Learning. Kelowna, British Columbia: Society for the Advancement of Excellence in Education.

Wang, T. H., Wang, K. H., Wang, W. L., Huang, S. C., and Chen, S. Y. (2004), Web-based Assistant and Test Analyses (WATA) System: Development and Evaluation, Journal of Computer Assisted Learning, 20(1): 59 – 71.

Wikipedia: E-assessment. Retrieved from http://en.m.wikipedia.org/wiki/E-assesment, 2015.

Zhenming, Y., Liang, Z., and Guohua, Z. (2003). A novel Web- Based online examination system for computer science education ", 33rd ASEE/IEEE Frontiers in Education Conference.S3F-7-S3F-10.

Appendix A
(Training Dataset)

Table A.1: 2466 records for training the developed Smart CBT model

S/N	Names	Phone	email	Time	Score	Class
1	Cairo	(684) 814-7246	quis.arcu@ametorciUt.edu	24	4	threat
2	Igor	(972) 405-7908	sem.mollis.dui@sitametconsectetuer.com	16	5	threat
3	Maris	(746) 650-4795	est@vitae.ca	15	1	threat
4	Urielle	(322) 252-4760	dui@ac.ca	24	5	threat
5	Desiree	(656) 370-1799	lobortis.ultrices@ornaretortorat.ca	22	10	no_threat
6	Samson	(353) 419-3269	eu@ipsumcursus.edu	12	7	medium_threat
7	Jakeem	(236) 416-8786	dui.lectus@etipsumcursus.co.uk	4	6	medium_threat
8	Maile	(347) 694-7845	Proin.eget@iaculisquispede.com	49	6	threat
9	Shad	(251) 692-8402	non@Donec.org	31	7	medium_threat
10	Priscilla	(169) 675-8576	nunc.nulla.vulputate@placerategetvenenatis.net	32	0	threat
11	Wing	(913) 164-6632	sit@eueuismod.com	28	7	medium_threat
12	Oscar	(623) 532-5373	non@placerategetvenenatis.co.uk	8	8	no_threat
13	Stephen	(574) 876-6704	odio@NullaaliquetProin.org	16	2	threat
14	Emma	(442) 782-2005	Phasellus.dapibus@suscipit.co.uk	25	0	threat
15	Abdul	(672) 715-6714	Cras.eget.nisi@etmagnis.net	4	10	no_threat
16	Ryder	(498) 549-8571	id.erat.Etiam@rutrumeu.com	50	3	threat
17	Phillip	(533) 864-2268	Donec.consectetuer.mauris@Nam.net	29	1	threat
18	Sydnee	(989) 319-4858	commodo@velitPellentesque.net	43	3	threat
19	Whoopi	(155) 989-9544	ac.mattis.ornare@sedturpisnec.com	7	4	threat
20	Martin	(230) 148-5980	eu.lacus@consectetuer.net	15	8	no_threat
21	Arsenio	(353) 144-7643	fermentum.convallis@sedduiFusce.net	50	1	threat
22	Caleb	(621) 639-9047	urna.suscipit@tristiquepharetra.co.uk	47	1	threat
23	Arsenio	(598) 573-1717	ipsum@consectetuermauris.ca	35	8	medium_threat
24	Robin	(939) 436-2307	sem.ut@vitaesodalesnisi.co.uk	28	3	threat
25	Savannah	(452) 506-8021	turpis@maurisblandit.edu	50	8	threat
26	Phyllis	(213) 927-0649	in.consequat.enim@sed.net	37	3	threat
27	Mannix	(394) 986-7799	taciti.sociosqu@aliquetmetus.co.uk	38	7	medium_threat
28	Naida	(618) 211-8743	pharetra.sed.hendrerit@Cras.co.uk	15	9	no_threat
29	Veronica	(652) 874-0326	condimentum@Nullamsuscipitest.com	29	9	medium_threat
30	Celeste	(281) 278-0441	ac@lacinia.org	19	6	medium_threat
31	Kaye	(291) 431-9251	in@non.edu	45	8	threat
32	Wanda	(878) 461-8720	in.faucibus.orci@lobortis.com	33	8	medium_threat
33	Kaye	(299) 973-6906	adipiscing@conubianostraper.ca	18	9	no_threat
34	Thor	(795) 696-4184	rhoncus.id.mollis@gravida.com	35	2	threat
35	Porter	(658) 212-0259	Phasellus.dapibus@dictumPhasellus.com	19	9	no_threat
36	Ainsley	(493) 280-5739	id@tristique.com	24	1	threat
37	Ronan	(418) 796-6122	euismod.in.dolor@posuere.com	28	2	threat

38	Ginger	(916) 717-2000	dis@loremsit.org	42	2	threat
39	Beck	(446) 719-8942	congue.turpis@amet.co.uk	4	1	threat
40	Fatima	(977) 347-1445	lacinia.orci.consectetuer@egestasascelerisque.net	26	1	threat
41	Wade	(821) 591-0430	pharetra.sed@Etiam.edu	20	6	medium_threat
42	Kiara	(299) 414-6925	sit.amet@metusIn.org	6	6	medium_threat
43	Abra	(334) 417-2061	neque.sed.dictum@congueturpisIn.co.uk	46	8	threat
44	Rhiannon	(500) 932-2824	non@laoreetliberoet.ca	47	3	threat
45	Darius	(982) 881-1970	ultrices.mauris.ipsum@amet.ca	21	2	threat
46	Mira	(452) 858-5528	est.congue@tempus.net	32	10	medium_threat
47	Hedy	(944) 889-9799	orci@eu.edu	20	6	medium_threat
48	Imogene	(748) 684-6763	netus.et.malesuada@nec.ca	35	10	medium_threat
49	Eugenia	(380) 331-3496	Maecenas.libero.est@liberolacusvarius.co.uk	41	3	threat
50	Paki	(807) 329-8136	iaculis@nibh.edu	4	9	no_threat
51	Matthew	(978) 705-9907	hendrerit@eteros.net	22	5	threat
52	Ralph	(714) 519-1596	Nulla.eget.metus@Aeneaneget.net	42	9	threat
53	Malachi	(757) 555-4405	sem@gravidaPraesent.edu	24	10	no_threat
54	Kylynn	(932) 296-7357	quis.massa@scelerisque.edu	50	6	threat
55	Howard	(703) 439-9844	sit@liberoDonecconsectetuer.com	39	1	threat
56	Chaim	(663) 485-0807	posuere.at.velit@parturientmontesnascetur.ca	37	3	threat
57	Giacomo	(217) 930-1232	vel.nisl@ametanteVivamus.com	30	10	medium_threat
58	Jade	(236) 421-4564	lacus@Donecnon.org	47	9	threat
59	Orli	(436) 662-3139	arcu.Vestibulum.ut@morbi.co.uk	40	2	threat
60	Elizabeth	(893) 620-2676	sed.libero@vehicula.ca	32	3	threat
61	Kaye	(908) 185-5368	vehicula@sagittisNullamvitae.ca	31	8	medium_threat
62	Levi	(707) 453-3536	Nunc@dui.edu	24	10	no_threat
63	Christopher	(990) 406-1098	morbi.tristique@etmagnis.org	48	2	threat
64	Zenaida	(353) 370-1836	lectus.sit@morbi.net	10	3	threat
65	Amir	(992) 252-3597	Sed@Naminterdumenim.org	5	7	medium_threat
66	Craig	(155) 333-6744	placerat@auctorvitae.com	4	5	threat
67	Leah	(541) 271-4118	placerat@Quisqueornare.co.uk	33	8	medium_threat
68	Zephr	(469) 332-9310	facilisis.facilisis@diamluctuslobortis.com	24	8	no_threat
69	Irma	(641) 545-8276	lacinia.at@imperdiet.com	46	3	threat
70	Clare	(756) 553-0242	Aenean.eget.metus@magnaUt.co.uk	45	2	threat
71	Philip	(332) 588-0308	tincidunt@enimEtiamimperdiet.ca	18	8	no_threat
72	Marah	(614) 223-9050	eget.ipsum.Donec@tristiquepharetraQuisque.org	11	3	threat
73	Katell	(936) 953-7275	ac.turpis.egestas@lectus.net	34	7	medium_threat
74	Daniel	(947) 966-3174	viverra@scelerisque.edu	29	4	threat
75	Knox	(162) 156-9670	Quisque.ornare@elit.co.uk	31	2	threat
76	Rafael	(395) 227-4707	scelerisque.mollis@arcu.ca	31	1	threat
77	Roth	(202) 843-6262	Cras.eget@odioapurus.co.uk	3	8	no_threat
78	Kyla	(135) 737-5131	libero@DonecegestasDuis.com	44	10	threat
79	Callum	(279) 212-6574	sem@sit.com	17	4	threat
80	Allen	(143) 658-2399	vitae@et.co.uk	40	1	threat

81	Randall	(850) 194-4311	mollis@dolor.org	16	4	threat
82	Zeus	(322) 511-1057	tempor.bibendum.Donec@vestibulum.ca	37	5	threat
83	Quinn	(269) 937-1562	ornare@Nullam.org	22	10	no_threat
84	Amena	(255) 204-1538	leo.elementum.sem@elementumloremut.com	36	10	medium_threat
85	Bryar	(122) 953-5279	risus@eueros.co.uk	42	3	threat
86	Xantha	(913) 365-9193	tincidunt.orci@porttitorinterdumSed.com	2	8	no_threat
87	Herrod	(430) 582-4468	nostra.per@Maecenasmalesuada.edu	17	4	threat
88	Aretha	(309) 537-0516	enim@velitPellentesque.net	21	2	threat
89	Carl	(200) 175-7965	mauris.Suspendisse.aliquet@lobortisquam.org	21	6	medium_threat
90	Matthew	(519) 297-0079	sapien.Nunc.pulvinar@insodales.ca	48	3	threat
91	Samson	(269) 846-7134	accumsan@sapiencursus.org	29	8	medium_threat
92	Logan	(395) 100-9225	fringilla.Donec.feugiat@adipiscing.net	44	7	threat
93	Bernard	(168) 316-4168	nunc.sit.amet@vitaealiquetnec.co.uk	21	4	threat
94	Mary	(772) 668-3963	facilisi.Sed.neque@Innec.co.uk	46	0	threat
95	Nina	(847) 858-3573	at.sem@eros.edu	48	10	threat
96	Remedios	(595) 726-5020	at@atnisiCum.ca	23	5	threat
97	Tanek	(632) 503-5053	facilisis.magna.tellus@cursus.org	42	5	threat
98	Dominique	(758) 511-6237	est.vitae@posuere.ca	45	1	threat
99	Elizabeth	(188) 306-3779	Integer.tincidunt@etmagnaPraesent.co.uk	26	2	threat
100	Anthony	(435) 124-5989	nunc@Aliquamfringillacursus.org	26	10	medium_threat
101	Echo	(783) 868-6598	ultrices.mauris.ipsum@a.co.uk	37	4	threat
102	Felix	(356) 821-8065	lorem.semper.auctor@egetlacus.net	2	6	medium_threat
103	Maryam	(857) 580-5119	leo@eueleifendnec.org	28	2	threat
104	Lisandra	(778) 984-4611	arcu.Vestibulum.ante@faucibus.org	9	2	threat
105	Velma	(928) 850-2113	sem.semper.erat@Mauris.com	18	5	threat
106	Kerry	(982) 992-4347	dis.parturient.montes@consectetueradipiscing.co.uk	27	4	threat
107	Cally	(281) 661-9816	habitant.morbi@Donecnon.com	46	7	threat
108	Elmo	(794) 794-5159	dolor.Donec.fringilla@fermentumarcu.co.uk	25	2	threat
109	Ria	(277) 386-7979	torquent.per.conubia@arcuVestibulum.ca	41	7	threat
110	Keane	(253) 925-0846	enim@orcitinciduntadipiscing.net	50	8	threat
111	Odysseus	(293) 678-0824	amet@enim.com	23	8	no_threat
112	Miranda	(525) 939-6101	ultrices.iaculis@justo.com	17	2	threat
113	Herman	(968) 964-6676	faucibus.leo.in@velitSed.com	24	5	threat
114	Summer	(781) 671-0353	In@etultricesposuere.net	5	6	medium_threat
115	Baker	(469) 249-9216	lorem.ac.risus@neque.com	34	3	threat
116	Shelby	(767) 198-4712	taciti.sociosqu@accumsaninterdum.com	45	10	threat
117	Zeph	(795) 918-2029	sodales.purus@dictumsapienAenean.co.uk	36	9	medium_threat
118	Caesar	(303) 556-4706	facilisis@arcuvel.co.uk	24	8	no_threat
119	Melodie	(907) 355-9518	tincidunt.tempus@mi.co.uk	20	2	threat
120	Serina	(675) 572-0577	mauris.elit@liberoduinec.org	34	4	threat
121	Hasad	(565) 838-2684	rutrum.non.hendrerit@sitamet.com	16	5	threat
122	Michelle	(571) 794-2977	non@Proinnon.net	27	9	medium_threat
123	Dennis	(226) 894-9931	ut.erat.Sed@pede.co.uk	31	3	threat

124	Colby	(586) 718-6690	interdum@sapiencursus.co.uk	5	3	threat
125	Quinlan	(560) 864-7115	mauris.sit.amet@arcuCurabitur.com	16	2	threat
126	Hector	(272) 878-5281	quis.diam@ametmassa.edu	18	8	no_threat
127	Lunea	(532) 883-4295	magna.Ut@etmagnis.org	50	1	threat
128	Shelley	(656) 957-4073	massa.Integer@lacusEtiambibendum.org	16	7	medium_threat
129	Ori	(375) 556-7701	ornare.egestas@Sedet.co.uk	6	9	no_threat
130	Jena	(919) 378-1769	vel.turpis.Aliquam@sitamet.co.uk	49	2	threat
131	Quyn	(359) 604-7820	neque.pellentesque@atauctorullamcorper.net	7	6	medium_threat
132	Ifeoma	(986) 737-6759	faucibus.orci.luctus@diamnuncullamcorper.com	45	7	threat
133	Tucker	(455) 122-7090	hendrerit.neque.In@Maecenaslibero.org	49	0	threat
134	Ori	(743) 260-3067	feugiat@egestasurna.com	41	0	threat
135	Jemima	(649) 267-2923	pede.nonummy.ut@imperdietdictum.com	38	10	medium_threat
136	Martha	(117) 866-2177	sed.dictum.eleifend@magnaCras.edu	33	8	medium_threat
137	Lois	(228) 672-6725	malesuada@seddolor.com	24	10	no_threat
138	Nathan	(486) 813-2599	aliquet@Crasvulputate.co.uk	21	1	threat
139	Felix	(353) 658-9618	nulla.Integer@etmagnisdis.com	15	0	threat
140	Melinda	(929) 728-6509	purus.Nullam@semegestasblandit.net	39	7	threat
141	Shoshana	(516) 827-8500	pede.Suspendisse@adipiscingelit.com	32	6	medium_threat
142	Ulysses	(704) 987-9970	facilisis.lorem@et.net	50	10	threat
143	Chase	(791) 770-3206	et.magnis.dis@maurisblandit.com	35	10	medium_threat
144	Wylie	(524) 134-7906	nisl.sem@arcueuodio.edu	13	10	no_threat
145	Nash	(422) 356-5488	faucibus@molestie.org	3	4	threat
146	Dieter	(882) 408-2542	montes.nascetur@antedictummi.org	28	10	medium_threat
147	Cody	(880) 544-4278	hendrerit@elit.net	31	4	threat
148	Ila	(681) 990-9356	libero.et.tristique@dapibusquamquis.org	33	6	medium_threat
149	Fulton	(386) 949-3830	urna@pede.co.uk	48	2	threat
150	James	(610) 915-9867	consectetuer.adipiscing@ante.net	32	5	threat
151	Judah	(310) 432-0186	adipiscing.enim@Aliquamvulputate.edu	9	1	threat
152	Maris	(522) 130-9546	pede.nonummy@nibh.com	48	9	threat
153	Mercedes	(719) 389-2385	molestie.Sed.id@orci.ca	45	6	threat
154	Fleur	(641) 253-9775	scelerisque.dui@dapibusligula.ca	41	3	threat
155	Amos	(180) 159-4484	congue@senectusetnetus.ca	9	8	no_threat
156	Robin	(904) 946-9966	tristique@ipsum.com	9	9	no_threat
157	Colette	(329) 430-4281	posuere.enim@Seddictum.edu	26	6	medium_threat
158	Jacob	(228) 932-8716	felis@sitametconsectetuer.ca	46	10	threat
159	Martha	(220) 250-6517	Duis@necimperdiet.org	46	4	threat
160	Justina	(180) 908-2140	elementum.sem@quisturpisvitae.com	45	2	threat
161	Gay	(264) 545-0479	adipiscing.ligula.Aenean@mauris.org	12	5	threat
162	Rogan	(547) 118-2267	orci.quis@et.org	24	8	no_threat
163	Noble	(224) 239-2289	semper.erat.in@quispedeSuspendisse.org	32	2	threat
164	Nichole	(255) 991-3061	erat.eget@idblanditat.com	13	1	threat
165	Priscilla	(181) 201-7375	hymenaeos.Mauris.ut@eu.ca	24	1	threat
166	Asher	(398) 388-3496	vel@orci.com	26	9	medium_threat

167	Elliott	(860) 467-3962	non@elitpedemalesuada.co.uk	5	3	threat
168	Edward	(333) 748-4857	ultrices.a@ornareFusce.ca	6	3	threat
169	Brock	(180) 117-0383	egestas.urna.justo@semper.net	24	8	no_threat
170	Julie	(383) 191-0887	mauris.sagittis.placerat@ipsum.ca	6	2	threat
171	Leo	(773) 778-0403	malesuada.fames.ac@Donecnonjusto.ca	1	8	no_threat
172	Jelani	(716) 757-5567	ut@egestasAliquamfringilla.com	20	6	medium_threat
173	Nehru	(203) 847-7428	viverra.Donec.tempus@vulputatevelit.co.uk	5	1	threat
174	Carl	(536) 888-7831	egestas@leoMorbineque.co.uk	22	0	threat
175	Ingrid	(377) 982-9901	placerat.orci@Sedpharetrafelis.com	12	0	threat
176	Ina	(731) 160-2727	In.mi@tortor.edu	2	10	no_threat
177	Peter	(921) 978-1116	Integer.id@dis.com	4	5	threat
178	Alan	(358) 468-8182	ante.dictum.cursus@Innecorci.co.uk	50	9	threat
179	Amos	(900) 313-0632	Aliquam.erat@lobortisrisusIn.com	5	8	no_threat
180	Hakeem	(814) 993-5741	Nulla@ametultricies.co.uk	2	10	no_threat
181	Petra	(331) 418-6325	et.arcu.imperdiet@enim.co.uk	34	10	medium_threat
182	Elmo	(188) 604-0212	molestie@lacinia.com	5	10	no_threat
183	Asher	(376) 232-2055	a.feugiat.tellus@pedesagittisaugue.com	17	9	no_threat
184	Avram	(518) 135-4756	Aliquam.nisl.Nulla@et.net	46	3	threat
185	Lee	(266) 914-9664	condimentum@convallis.com	18	0	threat
186	Chase	(781) 179-4129	a.aliquet@Fuscemilorem.co.uk	3	6	medium_threat
187	Blair	(898) 866-0618	leo@risusMorbimetus.net	38	9	medium_threat
188	Quail	(486) 961-7970	Cum.sociis.natoque@Morbi.ca	25	2	threat
189	Jermaine	(688) 965-2011	posuere@Donectemporest.org	7	0	threat
190	Shelly	(395) 217-0529	metus@diam.com	42	9	threat
191	Laith	(872) 680-6742	rhoncus.Proin.nisl@Suspendissecommodo.com	44	1	threat
192	Leigh	(381) 220-7974	nulla@arcuiaculisenim.org	16	6	medium_threat
193	Hu	(797) 594-6187	Vivamus@nisiCumsociis.edu	19	4	threat
194	Virginia	(832) 547-6578	Phasellus@magna.ca	5	10	no_threat
195	Hop	(251) 236-1882	sem.egestas@Cras.net	40	2	threat
196	Whoopi	(513) 255-3563	Nulla.aliquet@dictumsapienAenean.com	8	9	no_threat
197	Yael	(170) 345-6710	tellus.eu.augue@arcueu.org	39	2	threat
198	Kristen	(845) 321-9747	malesuada.augue@Vestibulumanteipsum.org	39	1	threat
199	Hammett	(898) 996-8114	rutrum.urna.nec@diam.net	12	8	no_threat
200	Cameran	(105) 626-8550	amet@nisl.com	46	8	threat

Appendix B
(Testing Dataset)

Table B.1: 617 records for testing the developed Smart CBT model

S/N	Name	Phone Number	Email Address	Time Score		Class
1	Larissa	(700) 114-8014	nonummy.ultricies.ornare@egestas.ca	3	9	no_threat
2	Imelda	(227) 127-9013	sed.pede@netuset.org	4	5	threat
3	Fay	(492) 549-2654	neque.Sed.eget@Maecenasornare.edu	47	9	threat
4	Destiny	(549) 938-0617	vel@lobortisrisusIn.co.uk	8	3	threat
5	Cassady	(469) 796-9298	laoreet.lectus@interdum.com	44	5	threat
6	Naomi	(499) 974-1752	quis.arcu.vel@famesac.com	27	9	medium_threat
7	Shana	(567) 114-1915	pede@commodohendreritDonec.com	37	6	medium_threat
8	Nasim	(626) 512-5482	Mauris@Donec.ca	24	8	no_threat
9	Vincent	(946) 994-5279	dui.lectus@nibh.co.uk	39	5	threat
10	Chaney	(723) 726-3880	massa.Quisque@urnaUt.net	49	1	threat
11	Carolyn	(980) 694-4376	vitae@ante.net	25	7	medium_threat
12	Mason	(137) 650-5171	lacus@ipsumSuspendisse.com	9	2	threat
13	Violet	(439) 844-2850	Quisque@aliquet.net	48	5	threat
14	Vaughan	(629) 324-4110	ad.litora.torquent@utmi.com	30	1	threat
15	Daryl	(711) 440-7077	nibh.Donec.est@lobortisquam.net	28	1	threat
16	Libby	(678) 137-3436	eu@nequevenenatislacus.edu	11	4	threat
17	Pascale	(298) 897-4006	consequat@tempor.com	16	3	threat
18	Hillary	(920) 305-8235	accumsan@facilisisnonbibendum.org	13	0	threat
19	Dolan	(665) 455-8570	dictum.cursus.Nunc@natoque.com	33	2	threat
20	Fletcher	(455) 368-1861	In.condimentum.Donec@ridiculusmus.ca	21	4	threat
21	Miriam	(254) 565-7929	dictum.ultricies.ligula@vitaevelitegestas.edu	50	1	threat
22	Cameran	(348) 262-1692	natoque@laoreet.edu	32	6	medium_threat
23	Ciaran	(869) 605-9279	facilisis@sedhendrerit.edu	45	6	threat
24	Erich	(880) 777-0197	magna.Lorem.ipsum@sodales.net	42	0	threat
25	Chandler	(275) 875-2512	Sed.congue@rutrumeu.org	13	8	no_threat
26	Tyrone	(488) 732-7148	lacus.Quisque@Crassedleo.net	50	5	threat
27	Nolan	(652) 207-0504	sem.ut.cursus@sodalesnisi.org	14	6	medium_threat
28	Jescie	(455) 390-1350	vel@convallisconvallis.ca	16	1	threat
29	Roary	(612) 187-9847	Integer@Maecenasornare.org	17	8	no_threat
30	Ignacia	(596) 191-2115	nibh@laoreetipsumCurabitur.co.uk	19	1	threat
31	Lareina	(930) 741-2125	a.mi.fringilla@enimnon.org	12	3	threat
32	Ivan	(192) 210-5772	habitant.morbi.tristique@mattisIntegereu.net	37	3	threat
33	Tashya	(158) 707-0606	euismod@Pellentesque.edu	50	10	threat

34	Brynn	(956) 887-5963	Aenean.euismod@fermentummetusAenean.edu	30	10	medium_threat
35	Alden	(379) 428-2298	eu.tempor@Uttincidunt.co.uk	43	1	threat
36	Carter	(249) 263-7057	vulputate@nectempus.net	18	0	threat
37	Phillip	(202) 745-3258	et.magnis@ligula.co.uk	27	2	threat
38	Yolanda	(786) 650-9126	et@nuncid.co.uk	23	0	threat
39	Ariel	(500) 184-9247	Curabitur.consequat.lectus@NullamenimSed.edu	12	2	threat
40	Rylee	(591) 440-3515	enim.condimentum.eget@feugiat.org	34	6	medium_threat
41	Seth	(663) 869-9911	Pellentesque@ipsumSuspendissenon.com	15	1	threat
42	Lyle	(477) 844-1562	dictum.Phasellus.in@Sedpharetrafelis.net	40	0	threat
43	Lyle	(297) 376-2596	ut.eros.non@erosNamconsequat.ca	49	8	threat
44	Libby	(293) 401-3131	ligula.Aenean@eratin.org	30	4	threat
45	Vera	(651) 264-3536	Duis@magnaDuisdignissim.net	11	0	threat
46	Shafira	(773) 340-2812	lectus.Cum.sociis@Nuncsed.net	7	5	threat
47	Doris	(348) 799-6672	pede@diameudolor.ca	36	8	medium_threat
48	Belle	(239) 664-9729	dapibus.rutrum.justo@Morbinon.edu	16	1	threat
49	Mari	(725) 668-2498	lacinia@sitamet.org	49	1	threat
50	Kenneth	(723) 343-9875	Pellentesque@semconsequatnec.com	30	9	medium_threat
51	Kareem	(579) 641-7131	faucibus.Morbi.vehicula@egestasblanditNam.org	30	1	threat
52	Kimberly	(117) 217-1394	quis.pede.Praesent@etmagnis.co.uk	8	10	no_threat
53	Ainsley	(886) 240-0338	sagittis@molestiepharetranibh.edu	29	7	medium_threat
54	Cole	(522) 782-0749	Nullam.nisl.Maecenas@justoProin.ca	41	5	threat
55	Ashton	(394) 117-1671	Nullam.lobortis@lorem.edu	6	10	no_threat
56	Stacey	(395) 778-9792	sem.ut@laciniavitaesodales.co.uk	28	8	medium_threat
57	Laura	(474) 213-0484	erat@aliquet.com	7	5	threat
58	Cora	(754) 509-0561	nulla@Nuncsollicitudin.com	19	1	threat
59	Madaline	(197) 534-4850	dapibus.id.blandit@feugiat.com	10	6	medium_threat
60	Leonard	(369) 620-6604	metus.Vivamus.euismod@Morbi.com	28	6	medium_threat
61	Uta	(424) 273-3538	vitae.diam@rhoncus.co.uk	42	6	threat
62	Rosalyn	(129) 808-0788	vel.faucibus.id@nislNullaeu.org	42	10	threat
63	Glenna	(173) 500-1652	semper@porttitor.net	30	1	threat
64	Rae	(663) 624-6554	enim@Morbi.com	3	5	threat
65	Eugenia	(397) 261-0107	arcu.imperdiet.ullamcorper@erateget.edu	33	10	medium_threat
66	Malcolm	(303) 597-0722	tincidunt@et.edu	49	8	threat
67	Ivana	(213) 600-4291	Curabitur@ametfaucibus.edu	31	9	medium_threat
68	Justin	(644) 985-7498	viverra.Donec@diamdictumsapien.org	40	10	threat
69	Maxwell	(175) 877-3748	lectus.ante@nonhendreritid.ca	22	8	no_threat
70	Zoe	(162) 311-4643	eleifend.non.dapibus@pretium.edu	39	7	threat
71	Shea	(547) 602-8483	arcu@utquam.co.uk	17	1	threat
72	James	(651) 510-5903	sociis@Suspendisse.com	48	0	threat
73	Karleigh	(997) 613-4439	ac.mattis@molestie.org	25	10	no_threat
74	Boris	(738) 544-3871	Integer@ligula.ca	3	0	threat
75	Cadman	(468) 506-5978	tincidunt.neque.vitae@leo.org	28	1	threat
76	Minerva	(265) 217-4533	Phasellus.ornare.Fusce@lorem.ca	14	0	threat

77	Jade	(651) 864-9345	auctor@non.ca	31	10	medium_threat
78	Sybill	(635) 174-3681	nisi@Vivamusnon.com	33	6	medium_threat
79	Ramona	(104) 469-6971	Nulla.eu@magna.org	43	5	threat
80	Donovan	(661) 754-3482	magna.Sed.eu@condimentumDonecat.co.uk	32	10	medium_threat
81	Amena	(157) 129-1409	nisi.dictum@primis.net	2	9	no_threat
82	Pamela	(963) 580-3002	mi.lacinia.mattis@risusvariusorci.net	13	1	threat
83	Britanney	(326) 935-0654	a.odio@metusIn.ca	46	9	threat
84	Richard	(531) 962-9441	cubilia.Curae.Donec@lectusrutrum.ca	35	8	medium_threat
85	Marah	(219) 806-0160	non@porttitor.com	8	4	threat
86	Malcolm	(766) 910-7780	Phasellus@sitamet.com	32	6	medium_threat
87	Aladdin	(782) 251-5164	mauris.blandit.mattis@adipiscingelit.edu	45	9	threat
88	Zelenia	(135) 440-6895	tempor.bibendum.Donec@nonenim.net	2	10	no_threat
89	Demetrius	(307) 411-7501	enim.nisl.elementum@nisimagnased.edu	24	3	threat
90	Rina	(570) 320-9640	Cras.interdum.Nunc@ultricies.ca	37	0	threat
91	Josiah	(203) 808-3216	Phasellus@enimgravida.com	8	1	threat
92	Sybill	(282) 580-9561	augue.Sed@nonmagna.org	19	9	no_threat
93	Nicholas	(732) 433-6211	id.erat@utipsum.net	43	3	threat
94	Blaze	(833) 539-5696	purus.in@Suspendisse.edu	27	10	medium_threat
95	Sasha	(253) 992-1676	volutpat@Integer.org	16	3	threat
96	Nissim	(429) 557-9113	et.eros.Proin@lectusconvallisest.net	23	5	threat
97	Vernon	(481) 832-1370	nibh.enim.gravida@ligula.edu	23	1	threat
98	Felicia	(670) 207-2110	urna.Nullam@eget.com	3	3	threat
99	Jocelyn	(337) 508-5714	elementum.at.egestas@Nam.net	17	9	no_threat
100	Avye	(632) 760-5820	consequat.purus@arcuVestibulumante.com	5	7	medium_threat
101	Dominic	(791) 756-8872	semper.erat@varius.ca	45	8	threat
102	Madeline	(505) 983-0970	venenatis.vel@VivamusrhoncusDonec.ca	50	10	threat
103	Uriel	(733) 168-1924	torquent.per@felisorci.org	45	1	threat
104	Hiroko	(515) 951-1337	et.risus@duiCraspellentesque.ca	45	9	threat
105	Elizabeth	(688) 498-7631	convallis.ante.lectus@aliquet.com	33	9	medium_threat
106	Jeremy	(543) 219-7301	lobortis@natoque.co.uk	45	1	threat
107	Rashad	(166) 117-0346	metus.Aenean@Sednulla.com	38	5	threat
108	Allegra	(404) 336-6117	vehicula@cursusdiamat.edu	1	4	threat
109	Tatum	(135) 947-8743	cursus.luctus.ipsum@tempuseuligula.net	21	5	threat
110	Barrett	(488) 883-7138	in@urna.co.uk	42	10	threat
111	Harlan	(452) 498-9238	Integer.urna.Vivamus@mollisduiin.net	34	5	threat
112	Lars	(667) 966-1149	et@nulla.net	37	0	threat
113	Penelope	(663) 293-8449	eu.tempor@sapienmolestieorci.edu	43	0	threat
114	Geoffrey	(704) 548-0863	velit.egestas.lacinia@Fusce.co.uk	19	9	no_threat
115	Mary	(664) 798-4195	mauris.sapien@etcommodoat.co.uk	18	9	no_threat
116	Yetta	(547) 817-2032	eget.laoreet.posuere@arcu.ca	17	0	threat
117	Sydney	(357) 333-0036	ipsum.ac@sedpedenec.com	42	0	threat
118	Connor	(395) 182-3595	eros.non.enim@aliquetnecimperdiet.co.uk	6	10	no_threat
119	Ursa	(900) 901-0393	ultrices.mauris@scelerisquemollisPhasellus.edu	8	9	no_threat

120	Gail	(234) 462-2392	pharetra@commodo.ca	35	9	medium_threat
121	4	(598) 429-6465	lacus@non.com	39	5	threat
122	Kenneth	(189) 489-5386	ipsum@gravidanuncsed.co.uk	50	10	threat
123	Leila	(526) 744-6487	arcu@orcilacus.org	42	4	threat
124	Beatrice	(882) 794-1733	non.dapibus@sitamet.com	1	1	threat
125	Benjamin	(572) 571-4558	dignissim.lacus.Aliquam@velconvallis.net	7	3	threat
126	Rama	(124) 568-1046	Sed.nulla.ante@elementumategestas.ca	44	4	threat
127	Harding	(795) 363-3552	dictum.ultricies.ligula@Aliquam.co.uk	27	9	medium_threat
128	Nina	(408) 877-5816	nisi.nibh@Namligulaelit.com	18	10	no_threat
129	Ashton	(854) 569-7495	egestas.rhoncus@musProin.org	35	4	threat
130	Dylan	(891) 524-2205	non@purus.net	3	7	medium_threat
131	Althea	(337) 884-0596	Quisque.porttitor.eros@consequatnec.com	5	0	threat
132	Alisa	(704) 119-6956	consectetuer.rhoncus.Nullam@nibhQuisque.com	43	10	threat
133	Ora	(386) 676-7259	libero.at.auctor@senectuset.com	3	6	medium_threat
134	Rhoda	(735) 233-0041	pede@netuset.net	2	4	threat
135	Abdul	(735) 989-6994	dolor.dolor@atnisi.ca	19	3	threat
136	Judith	(785) 380-2534	iaculis@congue.ca	35	3	threat
137	Sybill	(220) 881-6908	Morbi.sit.amet@Phasellus.net	34	0	threat
138	Adrienne	(685) 841-6012	Nunc@suscipit.com	32	0	threat
139	Jayme	(472) 673-0386	Donec@infaucibusorci.com	2	7	medium_threat
140	Jana	(138) 744-2948	eget.massa.Suspendisse@tinciduntduiaugue.ca	33	2	threat
141	Robert	(761) 123-4892	Aliquam.nec.enim@loremut.net	32	1	threat
142	Cedric	(362) 438-9594	ac@et.net	23	9	no_threat
143	Reece	(604) 181-8805	pharetra@Duisvolutpatnunc.co.uk	10	5	threat
144	Courtney	(464) 527-2478	erat.in.consectetuer@aaliquetvel.com	26	7	medium_threat
145	Maite	(132) 955-7936	Nullam.suscipit@malesuada.edu	24	2	threat
146	Quintessa	(754) 441-4094	dapibus@vel.edu	6	3	threat
147	Cara	(312) 229-0212	aliquet.Proin@ipsumnonarcu.net	24	8	no_threat
148	Mannix	(904) 232-7058	Sed@Craseutellus.ca	47	8	threat
149	Burke	(634) 959-8720	eget.metus.eu@ipsum.com	50	7	threat
150	Jarrod	(791) 739-4548	sed.facilisis@ultricesVivamusrhoncus.co.uk	45	5	threat
151	Alana	(750) 862-7410	a.ultricies@Nullamscelerisque.edu	28	5	threat
152	Matthew	(429) 544-3721	malesuada.ut@ligulaNullam.net	5	5	threat
153	David	(673) 741-1992	Maecenas.malesuada@dapibusligula.net	7	7	medium_threat
154	Tanek	(463) 205-2676	varius.orci.in@blanditcongue.net	45	1	threat
155	Cameran	(843) 101-7376	vulputate.risus@intempus.edu	22	1	threat
156	Lareina	(674) 763-1896	fringilla@orciluctus.edu	30	5	threat
157	Christian	(577) 794-3747	ante.lectus@metus.net	26	9	medium_threat
158	Giselle	(603) 652-7475	pede.Praesent@enimCurabitur.com	28	6	medium_threat
159	Dean	(460) 241-9415	dis.parturient.montes@adipiscingligula.com	43	10	threat
160	Duncan	(465) 318-3531	ante@pellentesquemassalobortis.ca	8	10	no_threat
161	Gannon	(820) 759-6379	erat@metuseu.edu	40	2	threat
162	Illiana	(652) 626-3946	In.condimentum.Donec@quisturpisvitae.ca	8	6	medium_threat

163	Malcolm	(983) 911-9086	dui.in.sodales@euplacerat.co.uk	7	9	no_threat
164	Anne	(942) 593-5635	Suspendisse.sagittis@Donecnibhenim.ca	29	4	threat
165	Kevin	(343) 900-3517	Phasellus.ornare@dapibusrutrumjusto.org	22	7	medium_threat
166	Paki	(281) 203-6184	Integer@eutelluseu.net	24	4	threat
167	Kelly	(374) 732-2912	placerat@tincidunt.edu	4	0	threat
168	Garth	(364) 445-5569	velit.in.aliquet@Fusce.co.uk	10	7	medium_threat
169	Adrian	(271) 132-8456	ultrices.mauris@turpis.org	40	0	threat
170	Kyle	(312) 336-4884	faucibus.Morbi@vestibulumMauris.net	25	5	threat
171	Abel	(888) 515-0017	pellentesque.a.facilisis@adipiscing.ca	5	7	medium_threat
172	Hall	(942) 130-9643	pellentesque@Maurisquisturpis.net	49	8	threat
173	Murphy	(278) 540-6769	adipiscing@egestas.edu	23	2	threat
174	Mufutau	(748) 371-3798	nec@arcu.co.uk	14	1	threat
175	Thane	(762) 953-9526	sit.amet.orci@duiquis.edu	39	10	threat
176	Rudyard	(401) 972-2400	ipsum.dolor.sit@mauriserat.com	26	2	threat
177	Brent	(994) 559-4522	hendrerit@DonecegestasAliquam.co.uk	7	3	threat
178	Callum	(357) 639-7392	non@at.edu	47	7	threat
179	Dexter	(221) 512-2176	ligula@massaSuspendisseeleifend.com	47	3	threat
180	Keegan	(378) 847-8202	odio@tinciduntorciquis.co.uk	7	5	threat
181	Micah	(865) 452-6643	ornare@sit.edu	4	0	threat
182	Hyatt	(491) 986-7473	sem.consequat@sed.com	29	5	threat
183	Dara	(488) 857-8136	lacus.Quisque@pulvinararcu.org	14	4	threat
184	Ivory	(508) 779-7053	facilisis.lorem@Nullam.net	30	1	threat
185	Cheryl	(392) 449-0975	lorem.auctor@malesuada.com	33	4	threat
186	Abel	(313) 212-6885	tempor.augue@ligulaNullam.com	9	0	threat
187	Arsenio	(155) 700-1973	ac@vel.ca	23	7	medium_threat
188	Tate	(153) 742-7034	Integer.vulputate.risus@pedemalesuadavel.co.uk	17	2	threat
189	Zeus	(545) 890-1785	adipiscing.elit.Curabitur@euismodacfermentum.com	41	2	threat
190	Lisandra	(436) 221-3162	vel.nisl.Quisque@urnaet.ca	12	9	no_threat
191	Quinn	(390) 886-5367	sapien.Cras@quis.edu	22	9	no_threat
192	Leila	(328) 339-1150	amet.dapibus.id@odioauctorvitae.net	27	7	medium_threat
193	Hayes	(732) 237-7551	facilisis@non.co.uk	45	3	threat
194	Fay	(502) 300-6545	ipsum.Donec@malesuadamalesuada.co.uk	39	9	threat
195	Lynn	(461) 273-1105	neque@doloregestas.edu	49	3	threat
196	Signe	(665) 104-5459	Duis.cursus@augueeu.edu	3	6	medium_threat
197	Luke	(818) 805-3384	aliquet@Donec.net	29	7	medium_threat
198	Noah	(630) 934-0120	hendrerit.id.ante@mi.com	44	6	threat
199	Cheryl	(808) 988-9097	pede.et@ultricesposuere.net	36	5	threat
200	Lisandra	(402) 928-4535	eleifend@montesnasceturridiculus.org	38	9	medium_threat
201	Ava	(738) 117-8754	Nam@nisiMauris.com	22	10	no_threat
202	Darryl	(134) 542-7315	Donec.felis@senectuset.net	28	2	threat
203	Dorothy	(975) 733-3106	tellus.sem.mollis@tempusnon.net	15	10	no_threat
204	Mannix	(578) 824-1787	placerat.orci@tristique.com	11	6	medium_threat
205	Paul	(248) 519-2033	magnis.dis.parturient@Cras.edu	33	4	threat

206	Raymond	(367) 453-9618	gravida.sit@quisurna.org	2	2	threat
207	Ainsley	(364) 455-1419	adipiscing@nec.ca	23	7	medium_threat
208	Oren	(221) 339-3128	eget@tellus.com	5	1	threat
209	Gray	(810) 962-8096	purus.ac@posuere.ca	27	6	medium_threat
210	Kylie	(508) 270-0973	sed.turpis.nec@risusQuisquelibero.net	22	9	no_threat
211	Odysseus	(649) 936-5422	justo.Praesent.luctus@ornare.ca	29	10	medium_threat
212	Jacqueline	(447) 483-1518	lorem.ut@Aliquamornare.ca	44	0	threat
213	Blair	(189) 488-7825	tortor.dictum.eu@non.net	4	6	medium_threat
214	Michelle	(555) 286-9828	auctor@urna.org	29	5	threat
215	Helen	(595) 225-7565	tellus.non.magna@In.edu	34	2	threat
216	Rhona	(946) 758-4599	Cras.eget@semper.net	39	6	threat
217	Wyoming	(566) 738-4299	ipsum.leo@feugiatnecdiam.net	24	2	threat
218	Chelsea	(915) 505-7778	facilisis@pedeac.com	6	6	medium_threat
219	Sylvester	(852) 710-9019	at.auctor.ullamcorper@orciDonec.org	47	10	threat
220	Donna	(122) 571-1493	ante@nisiCum.ca	39	8	threat
221	Iona	(520) 574-8612	Donec@vestibulumloremsit.co.uk	19	10	no_threat
222	Kiona	(713) 298-8472	interdum.libero@vulputateullamcorpermagna.co.uk	43	0	threat
223	Judah	(606) 217-4784	arcu.Sed.et@Sed.net	44	4	threat
224	Tashya	(517) 299-8111	suscipit.nonummy.Fusce@laciniavitae.com	47	9	threat

Appendix C
(Program Source Code Listing)

#Database Connection Code

```
<?php
//Database Connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname= "db";

$conn = mysqli_connect($servername, $username, $password,$dbname);

if (!$conn) {
   die("Connection failed: " . $conn->connect_error);
}
//echo "Connected successfully";

 ?>

#login Connection Code
<?php
ob_start();
session_start();

 include 'connection.php';

 if(isset($_POST['login'])){
  $email = mysqli_escape_string($conn,$_POST['email']);
  $pass = mysqli_escape_string($conn,$_POST['password']);

  $query1 = mysqli_query($conn,"SELECT * FROM admin_table WHERE email_address = '$email'");
  $row  = mysqli_fetch_assoc($query1);
  $status = $row['status'];
  $salt =   $row['salt'];
  $password = hash('sha256',$salt.$pass);

  $query = mysqli_query($conn,"SELECT * FROM admin_table WHERE email_address = '$email' and password = '$password' AND status =
'active'");
  $num = mysqli_num_rows($query);
  if($num == 1){
    $rows = mysqli_fetch_assoc($query);
    $fullname = $rows['firstname'].' '.$rows['lastname'];
    $admin_id  = $rows['admin_id'];
    $status = $rows['status'];
    $access = "Administrator";
    $datetime = date('Y-m-d H:i:s');
   $log = mysqli_query($conn,"INSERT INTO user_logs(admin_id,fullname,access_level,status,datetime)
          VALUES('$admin_id',
                '$fullname',
                 '$access',
                  '$status',
                 '$datetime')");

   $_SESSION['valid'] = true;
   $_SESSION['timeout'] = time();
```

```
    $_SESSION['email'] = $_POST['email'];
    header('location:../view/dashboard.php');
  }
  else{
    echo 'invalid';
  }

 }

 ?>


#Index Page
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <title> Smart CBT Login</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta content="Admin Dashboard " name="description" />
    <meta content="Themesbrand" name="author" />
    <!-- App favicon -->
    <link rel="shortcut icon" href="assets/images/favicon.ico">

    <!-- Bootstrap Css -->
    <link href="assets/css/bootstrap.min.css" id="bootstrap-style" rel="stylesheet" type="text/css" />
    <!-- Icons Css -->
    <link href="assets/css/icons.min.css" rel="stylesheet" type="text/css" />
    <!-- App Css-->
    <link href="assets/css/app.min.css" id="app-style" rel="stylesheet" type="text/css" />

</head>

<body>

    <div class="account-pages my-5 pt-sm-5">
        <div class="container">
            <div class="row justify-content-center">
                <div class="col-md-8 col-lg-6 col-xl-5">
                    <div class="card overflow-hidden">
                        <div class="bg-login text-center">
                            <div class="bg-login-overlay"></div>
                            <div class="position-relative">
                                <h5 class="text-white font-size-20">Welcome Back !</h5>
                                <p class="text-white-50 mb-0">Sign in to continue to Smart CBT.</p>
                                <a href="index.php" class="logo logo-admin mt-4">
                                    <img src="assets/images/logo-sm-dark.png" alt="" height="30">
                                </a>
                            </div>
                        </div>
                        <div class="card-body pt-5">
                            <div class="p-2">
                                <form class="form-horizontal" action="controller/login.php" method="post">

                                    <div class="form-group">
                                        <label for="username">Email</label>
                                        <input type="email" name = "email" class="form-control" required id="email" placeholder="Enter Email">
                                    </div>

                                    <div class="form-group">
                                        <label for="userpassword">Password</label>
                                        <input type="password" name = "password" class="form-control" required id="userpassword" placeholder="Enter
password">
                                    </div>


                                    <div class="mt-3">
                                        <button class="btn btn-primary btn-block waves-effect waves-light" name ="login" type="submit">Log In</button>
                                    </div>

                                </form>
                            </div>
```

```
                    </div>
                </div>
                <div class="mt-5 text-center">

                    <p>© 2020 Smart CBT. Crafted with <i class="mdi mdi-heart text-danger"></i> by Beautifulcode Inc</p>
                </div>

            </div>
        </div>
    </div>
</div>

<!-- JAVASCRIPT -->
<script src="assets/libs/jquery/jquery.min.js"></script>
<script src="assets/libs/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="assets/libs/metismenu/metisMenu.min.js"></script>
<script src="assets/libs/simplebar/simplebar.min.js"></script>
<script src="assets/libs/node-waves/waves.min.js"></script>

<script src="assets/js/app.js"></script>

</body>


</html>
```

#Controller Logic Script

```
<?php


include 'connection.php';
include 'connection.php';
require_once('vendor/php-excel-reader/excel_reader2.php');
require_once('vendor/SpreadsheetReader.php');

session_start();
if (!isset($_SESSION['email']))
{
   header("Location: ../index.php");
   die();
}

$log_id = "";

$email_id  = $_SESSION['email'];
$n = mysqli_query($conn,"SELECT * FROM admin_table");
$r = mysqli_fetch_assoc($n);
$a_id  = $r['admin_id'];
$current_datetime = date("Y-m-d") . ' ' . date("H:i:s", STRTOTIME(date('h:i:sa')));

//random string generator
function generate_string($input, $strength = 16) {
   $input_length = strlen($input);
   $random_string = '';
   for($i = 0; $i < $strength; $i++) {
      $random_character = $input[mt_rand(0, $input_length - 1)];
      $random_string .= $random_character;
   }

   return $random_string;
}

//register CBT admin
if (isset($_POST['register_admin']) && !empty($_FILES["file"]["name"])){

  $fname = mysqli_escape_string($conn,$_POST['firstname']);
  $lname = mysqli_escape_string($conn,$_POST['lastname']);
  $email = mysqli_escape_string($conn,$_POST['email']);
  $phone = mysqli_escape_string($conn,$_POST['phone']);
  //encrpt password using (Secure Hash Algorithm 256 bit)
  //step1 grab the password text
  $pass = mysqli_escape_string($conn,$_POST['password']);
  //encrypt with sha256 with salt
```

```
$permitted_chars = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
$salt = generate_string($permitted_chars, 100);
$password = hash('sha256',$salt.$pass);

$created_on =  date('Y-m-d H:i:s');
//copy the file into the new path
$statusMsg = '';// File upload path
$targetDir = "../images_admin/";
$fileName = basename($_FILES["file"]["name"]);
$targetFilePath = $targetDir . $fileName;
$fileType = pathinfo($targetFilePath,PATHINFO_EXTENSION);

$allowTypes = array('jpg','png','jpeg','gif','pdf');
   if(in_array($fileType, $allowTypes)){
     //check if email already exist
     $check_email = mysqli_query($conn,"SELECT * FROM admin_table  WHERE email_address = '$email' ");
     $e  = mysqli_num_rows($check_email);
     if ($e > 1) {
         echo "email ready exists";
     }
     else {
       //check if phone already exist
       $check_email = mysqli_query($conn,"SELECT * FROM admin_table  WHERE phone_number = '$phone' ");
       $ph  = mysqli_num_rows($check_email);
       if ($ph > 1) {
           echo "phone number exist";



       }
       else{
         // Upload file to server
         if(move_uploaded_file($_FILES["file"]["tmp_name"], $targetFilePath)){
             // Insert image file name into database
             $query = "INSERT INTO
admin_table(firstname,lastname,phone_number,email_address,password,admin_created_on,picture,status,salt)
                                          VALUES('$fname',
                                              '$lname',
                                              '$phone',
                                              '$email',
                                              '$password',
                                              '$created_on',
                                              '$fileName',
                                              'active',
                                               '$salt'
                                             )";
             $execute = mysqli_query($conn,$query);
             if($execute){
               echo "saved";
             }
             else{
               echo "Not saved";
             }


           }
         }

       }
     }
   }

//add exams

if(isset($_POST['AddExam'])){
     //get admin // Id
     $getid = mysqli_fetch_assoc(mysqli_query($conn,"SELECT * FROM admin_table WHERE email_address = '$email_id'"));
     $admin_id = $getid['admin_id'];
     $online_exam_title  =mysqli_escape_string($conn,$_POST['online_exam_title']);
     $online_exam_datetime = mysqli_escape_string($conn,$_POST['online_exam_datetime'] . ':00');
     $online_exam_duration = mysqli_escape_string($conn,$_POST['online_exam_duration']);
     $total_question = mysqli_escape_string($conn,$_POST['total_question']);
     $marks_per_right_answer = mysqli_escape_string($conn, $_POST['marks_per_right_answer']);
     $marks_per_wrong_answer = mysqli_escape_string($conn,$_POST['marks_per_wrong_answer']);
```

```
        $online_exam_created_on = $current_datetime;
        $online_exam_status = 'Pending';
        $online_exam_code   = md5(rand());

    $query = "INSERT INTO exam_table
      (admin_id,exam_title,exam_datetime,exam_duration, total_question, marks_per_right_answer,
marks_per_wrong_answer,exam_created_on,exam_status, exam_code)
      VALUES
      ('$admin_id', '$online_exam_title',
       '$online_exam_datetime', '$online_exam_duration',
      '$total_question', '$marks_per_right_answer',
      '$marks_per_wrong_answer', '$online_exam_created_on',
      '$online_exam_status', '$online_exam_code')
      ";
    $add = mysqli_query($conn,$query);
    if($add){
        header('location:../view/add_exam.php');

    }



}

//send notification

if(isset($_POST['send'])){

  $title = mysqli_escape_string($conn,$_POST['title']);
  $exam_id = mysqli_escape_string($conn,$_POST['exam_id']);
  $message = mysqli_escape_string($conn,$_POST['message']);

  $status = "sent";
  $time = date("Y-m-d") . ' ' . date("H:i:s", STRTOTIME(date('h:i:sa')));

  //insert into SQLiteDatabase

  $query = "INSERT INTO send_notification(title,exam_id,message,status,created_on,admin_id)VALUES(
    '$title',
    '$exam_id',
    '$message',
    '$status',
    '$time',
    '$a_id'
  )";

  $execute_query =mysqli_query($conn,$query);
  if($execute_query) {
    echo 'saved';
  }
  else{
    echo 'not saved';
  }

}

//register Students
if (isset($_POST['register_student']) && !empty($_FILES["file"]["name"])){

  $fname = mysqli_escape_string($conn,$_POST['firstname']);
  $lname = mysqli_escape_string($conn,$_POST['lastname']);
  $mname = mysqli_escape_string($conn,$_POST['middlename']);
  $dob = mysqli_escape_string($conn,$_POST['dob']);
  $email = mysqli_escape_string($conn,$_POST['email']);
  $phone = mysqli_escape_string($conn,$_POST['phone']);
  $state = mysqli_escape_string($conn,$_POST['state']);
  $lga = mysqli_escape_string($conn,$_POST['lga']);
  $gender = mysqli_escape_string($conn,$_POST['gender']);
  $mmname = mysqli_escape_string($conn,$_POST['mother_maiden_name']);
  $petname = mysqli_escape_string($conn,$_POST['petname']);
  $favfood = mysqli_escape_string($conn,$_POST['favfood']);
  $favsport = mysqli_escape_string($conn,$_POST['favsport']);
  $town = mysqli_escape_string($conn,$_POST['town']);
  //encrpt password using (Secure Hash Algorithm 256 bit)
```

```
 //step1 grab the password text
 $pass = mysqli_escape_string($conn,$_POST['password']);
 //encrypt with sha256 with salt
 $permitted_chars = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
 $salt = generate_string($permitted_chars, 100);
 $password = hash('sha256',$salt.$pass);

 $created_on =  date('Y-m-d H:i:s');
 //copy the file into the new path
 $statusMsg = '';// File upload path
 $targetDir = "../images_admin/";
 $fileName = basename($_FILES["file"]["name"]);
 $targetFilePath = $targetDir . $fileName;
 $fileType = pathinfo($targetFilePath,PATHINFO_EXTENSION);

 $allowTypes = array('jpg','png','jpeg','gif','pdf');
    if(in_array($fileType, $allowTypes)){
      //check if email already exist
      $check_email = mysqli_query($conn,"SELECT * FROM student_table  WHERE email = '$email' ");
      $e  = mysqli_num_rows($check_email);
      if ($e > 1) {
          echo "email ready exists";
      }
      else {
        //check if phone already exist
        $check_email = mysqli_query($conn,"SELECT * FROM student_table  WHERE phone = '$phone' ");
        $ph  = mysqli_num_rows($check_email);
        if ($ph > 1) {
            echo "phone number exist";

        }
        else{
          // Upload file to server
          if(move_uploaded_file($_FILES["file"]["tmp_name"], $targetFilePath)){
              // Insert image file name into database
              $query = "INSERT INTO
student_table(firstname,lastname,middlename,email,password,phone,mother_maiden_name,user_image,favoritesport,petname,favoritefood,state,l
ga,dob,town,salt,user_created_on)
                                          VALUES('$fname',
                                              '$lname',
                                              '$mname',
                                              '$email',
                                              '$password',
                                              '$phone',
                                              '$mmname',
                                              '$fileName',
                                              '$favsport',
                                              '$petname',
                                              '$favfood',
                                              '$state',
                                              '$lga',
                                              '$dob',
                                              '$town',
                                               '$salt',
                                               '$created_on'
                                             )";
              $execute = mysqli_query($conn,$query);
              if($execute){
                echo "saved";
              }
              else{
                echo "Not saved";
              }


            }
          }

        }
      }
      }


//change password
```

```
if(isset($_POST['changepassword'])){

  $email = mysqli_escape_string($conn,$_POST['email']);
  $pass = mysqli_escape_string($conn,$_POST['password']);
  $cpassword = mysqli_escape_string($conn,$_POST['cpassword']);

  if($pass != $cpassword) {
    echo "password mismatch";

  }
  else {
      //update admin $password

      //encrypt with sha256 with salt
      $permitted_chars = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
      $salt = generate_string($permitted_chars, 100);
      $password = hash('sha256',$salt.$pass);

      $update = mysqli_query($conn,"UPDATE admin_table SET password = '$password', salt = '$salt' WHERE email_address = '$email'");
        if ($update){
          echo 'password changed';
        }
        else{
          echo 'failed';
        }

  }

}

//update student Info

if (isset($_POST['update_student']) ){

   $sid = mysqli_escape_string($conn,$_POST['sid']);
  $dob = mysqli_escape_string($conn,$_POST['dob']);
  $email = mysqli_escape_string($conn,$_POST['email']);
  $phone = mysqli_escape_string($conn,$_POST['phone']);

  $petname = mysqli_escape_string($conn,$_POST['petname']);
  $favfood = mysqli_escape_string($conn,$_POST['favfood']);
  $favsport = mysqli_escape_string($conn,$_POST['favsport']);
  $town = mysqli_escape_string($conn,$_POST['town']);
  //encrpt password using (Secure Hash Algorithm 256 bit)
  //step1 grab the password text
  $pass = mysqli_escape_string($conn,$_POST['password']);
  //encrypt with sha256 with salt
  $permitted_chars = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
  $salt = generate_string($permitted_chars, 100);
  $password = hash('sha256',$salt.$pass);



      //check if email already exist
      $check_email = mysqli_query($conn,"SELECT * FROM student_table  WHERE email = '$email' ");
      $e  = mysqli_num_rows($check_email);
      if ($e > 1) {
          echo "email ready exists";
      }
      else {
        //check if phone already exist
        $check_email = mysqli_query($conn,"SELECT * FROM student_table  WHERE phone = '$phone' ");
        $ph  = mysqli_num_rows($check_email);
        if ($ph > 1) {
            echo "phone number exist";

        }
        else{

            // Insert image file name into database
            $query = "UPDATE student_table SET email = '$email',
                                    password = '$password',
                                    phone = '$phone',
```

```
                                favoritesport = '$favsport',
                                petname = '$petname',
                                favoritefood = '$favfood',
                                dob = '$dob',
                                town = '$town',
                                salt = '$salt'
                                WHERE student_id = '$sid'";
            $execute = mysqli_query($conn,$query);
            if($execute){
              echo "saved";
            }
            else{
              echo "Not saved";
            }


          }
        }

      }

// deactivate users
/*$log_id = $_GET['log_id'];
if($_GET['log_id'] != NULL ){

  $query = mysqli_query($conn,"UPDATE admin_table SET status  = 'Inactive' WHERE email_address = '$log_id' ");

  header('location:../view/Dashboard.php');
}
*/

if (isset($_POST["import"]))
{

  $allowedFileType = ['application/vnd.ms-excel','text/xls','text/xlsx','application/vnd.openxmlformats-officedocument.spreadsheetml.sheet'];

  if(in_array($_FILES["file"]["type"],$allowedFileType)){

      $targetPath = 'uploads/'.$_FILES['file']['name'];
      move_uploaded_file($_FILES['file']['tmp_name'], $targetPath);

      $Reader = new SpreadsheetReader($targetPath);

      $sheetCount = count($Reader->sheets());
      for($i=0;$i<$sheetCount;$i++)
      {

        $Reader->ChangeSheet($i);

        foreach ($Reader as $Row)
        {

          $question_id = "";
          if(isset($Row[0])) {
            $question_id = mysqli_real_escape_string($conn,$Row[0]);
          }

          $online_exam_id = "";
          if(isset($Row[1])) {
            $online_exam_id = mysqli_real_escape_string($conn,$Row[1]);
          }
          $question_title= "";
          if(isset($Row[2])) {
            $question_title = mysqli_real_escape_string($conn,$Row[2]);
          }

          $answer_option = "";
          if(isset($Row[3])) {
            $answer_option = mysqli_real_escape_string($conn,$Row[3]);
          }
          $option_1 = "";
          if(isset($Row[4])) {
            $option_1 = mysqli_real_escape_string($conn,$Row[4]);
          }
```

```
            $option_2 = "";
            if(isset($Row[5])) {
               $option_2 = mysqli_real_escape_string($conn,$Row[5]);
            }
            $option_3 = "";
            if(isset($Row[6])) {
               $option_3 = mysqli_real_escape_string($conn,$Row[6]);
            }
            $option_4 = "";
            if(isset($Row[7])) {
               $option_4 = mysqli_real_escape_string($conn,$Row[7]);
            }
          //AES encryption scheme
          $cipher = "AES-128-CTR";
          //Open ssl encryption method
          $ivlen = openssl_cipher_iv_length($cipher);
          $option = 0;
          //non null initialization of the vector for encryption.
          $encryption_iv = "9028993933845222";
          //store encrytion key => ,
          $encryption_key =  "h5hd7jdjds9k3jjdkd9kdkejdi93jkjdp1";
          $question_title = openssl_encrypt($question_title,$cipher,$encryption_key,$option,$encryption_iv);
          $answer_option = openssl_encrypt($answer_option,$cipher,$encryption_key,$option,$encryption_iv);
          $option_1 = openssl_encrypt($option_1,$cipher,$encryption_key,$option,$encryption_iv);
          $option_2 = openssl_encrypt($option_2,$cipher,$encryption_key,$option,$encryption_iv);
          $option_3 = openssl_encrypt($option_3,$cipher,$encryption_key,$option,$encryption_iv);
          $option_4 = openssl_encrypt($option_4,$cipher,$encryption_key,$option,$encryption_iv);



            if (!empty($question_id) || !empty($exam_id)) {
               $query = "insert into question_table(question_id,online_exam_id,question_title,option_1,option_2,option_3,option_4,answer_option)
               values('".$question_id."',
                    '".$online_exam_id."',
                    '".$question_title."',
                    '".$option_1."',
                    '".$option_2."',
                    '".$option_3."',
                    '".$option_4."',
                    '".$answer_option."'
      )";
               $result = mysqli_query($conn, $query);

               if (! empty($result)) {
                  $type = "success";
                  $message = "Excel Data Imported into the Database";
                  header('location:../view/add_questions.php');
               } else {
                  $type = "error";
                  $message = "Problem in Importing Excel Data";
               }
            }
         }

      }
  }
  else
  {
      $type = "error";
      $message = "Invalid File Type. Upload Excel File.";
  }
}

//updateexams


?>


#Exam Page
<!DOCTYPE html>
<html lang = "en">
```

```
 <link rel="stylesheet" href="inc/TimeCircles.css" />

<?php
  include ('examheader.php');

?>

    <link rel="stylesheet" href="inc/TimeCircles.css" />

<?php
//session_start();
extract($_POST);
include("connect.php");

if (isset($_POST['send'])){

  $exam_id = $_POST['exam_id'];
  $f = "SELECT * FROM exam_table WHERE exam_id = '$email'";
  $s  = mysqli_query($conn,$f);
  $row = mysqli_fetch_assoc($s);
  $examcode = $row['exam_code'];
  $_SESSION['exam_id'] = $exam_id;

  $result = mysqli_query($conn,"SELECT * FROM exam_table WHERE exam_id = '$exam_id'");

  $row = mysqli_fetch_assoc($result);

  $exam_status = $row['exam_status'];
  $exam_title = $row['exam_title'];
  $exam_star_time = $row['exam_datetime'];
  $duration = $row['exam_duration'] . ' minutes';
  $time =  $row['exam_duration'] * 60 ;
  $exam_end_time = strtotime($exam_star_time . '+' . $duration);
  $total = $row['total_question'];

  $exam_end_time = date('Y-m-d H:i:s', $exam_end_time);
  $remaining_minutes = strtotime($exam_end_time) - time();
  //get time from time db

}

$_SESSION['total'] =   $total;

foreach ($_SESSION as $key => $value) {
  $key = $value;
}




//unset($_SESSION['qa']);
if(!isset($_SESSION['qa']) || empty($_SESSION['qa'])){
  $q = "select * from question_table where online_exam_id = '".$exam_id."' order by RAND() limit 50";
   $res= mysqli_query($conn,$q);
   if(mysqli_num_rows($res)<=0)

   {
    ?>
    <script>
       alert('Sorry the questions are not available. Try with some other subject');
       window.location='exam.php';
     </script>
    <?php
   }

  $_SESSION['qa'] = array();
  $count = 1;
  while($row = mysqli_fetch_assoc($res)){

    $id = $row['question_id'];
    $question = $row['question_title'];
    $a = $row['option_1'];
```

```
  $b = $row['option_2'];
  $c = $row['option_3'];
  $d = $row['option_4'];

  //AES encryption scheme
  $cipher = "AES-128-CTR";
  //Open ssl encryption method
  $ivlen = openssl_cipher_iv_length($cipher);
  $option = 0;
  //non null initialization of the vector for encryption.
  $encryption_iv = "9028993933845222";
  //store encrytion key => ,
  $encryption_key =  "h5hd7jdjds9k3jjdkd9kdkejdi93jkjdp1";
  $question_title = openssl_decrypt($question,$cipher,$encryption_key,$option,$encryption_iv);
  //$answer_option = openssl_decrypt($,$cipher,$encryption_key,$option,$encryption_iv);
  $option_1 = openssl_decrypt($a,$cipher,$encryption_key,$option,$encryption_iv);
  $option_2 = openssl_decrypt($b,$cipher,$encryption_key,$option,$encryption_iv);
  $option_3 = openssl_decrypt($c,$cipher,$encryption_key,$option,$encryption_iv);
  $option_4 = openssl_decrypt($d,$cipher,$encryption_key,$option,$encryption_iv);

  $_SESSION['qa'][$count]['id'] = $row['question_id'];
  $_SESSION['qa'][$count]['question'] = $question_title;
  $_SESSION['qa'][$count]['option_a'] = $option_1;
  $_SESSION['qa'][$count]['option_b'] = $option_2;
  $_SESSION['qa'][$count]['option_c'] = $option_3;
  $_SESSION['qa'][$count]['option_d'] = $option_4;
  $_SESSION['qa'][$count]['ans'] = '';
  $count++;
 }



}

//echo $_SESSION['qa'][1]['question'];exit();

?>

<body onkeydown="return (event.keyCode == 154)">

<div class="main-content">

        <div class="page-content">

           <!-- start page title -->
           <div class="row">
              <div class="col-12">
                 <div class="page-title-box d-flex align-items-center justify-content-between">
                    <h4 class="page-title mb-0 font-size-18">Course : <?php echo $exam_title; ?> Exam</h4>  
                    <h4 class="page-title mb-0 font-size-18">No. of Question : <?php echo $total; ?> </h4> 
                     <h4 class="page-title mb-0 font-size-18">Duration : <?php echo $duration; ?> </h4>
                     <h4 class="page-title mb-0 font-size-18">Status :  In progress </h4>


                 </div>
              </div>
           </div>
           <!-- end page title -->

           <div class="row">
              <div class="col-lg-8">

                 <div class="card">
                    <div class="card-body">
        <a href="result_count.php" id="endExamBtn" class="btn btn-primary"> End-Exam</a>



                      <iframe id="exam_frame" width="100%" height="450" frameborder="0" style="background-color:white;" scrolling="no"
src="exam_frame.php"></iframe>


                 </div>
```

```
                    </div>
                  </div>
                  <!-- end card -->

                <!-- end col -->

                <div class="col-lg-4">
                  <div class="card">
                    <div class="card-body">
                      <center>   <h4 class="card-title">Personal Information </h4></center>
                      <table>
                       <tr>
                        <td rowspan="2">
                          <div class="text-center" dir="ltr">
                            <img class="rounded-circle header-profile-user" style="width:100px; height:100px;" src="<?php echo
'../../images_admin/'.$img; ?>" alt="Header Avatar">

                          </div>
                        </td>

                        <td>
                          <span class="d-none d-xl-inline-block ml-1">First Name : <?php echo $fname; ?></span>

                        </td>


                        <td>
                          <span class="d-none d-xl-inline-block ml-1">Last Name : <?php echo $lname; ?></span>
                        </td>

                       </tr>

                      </table>

                      <center>   <h4 class="card-title">Examination Clock </h4></center>

                      <div align="center">
                        <div id="CountDownTimer" data-timer="<?php echo $time; ?>"  style="max-width:400px; width: 100%; height:
200px;"></div>
                      </div>

                    </div>
                  </div>
                  <!-- end card -->
                </div>
                <!-- end col -->
              </div>


          </div>
          <!-- End Page-content -->

        </div>

        <!-- end main content-->
      </body>

<?php

include('footer.php');

?>

<script type="text/javascript" src="inc/jquery.min.js"></script>
    <script type="text/javascript" src="inc/TimeCircles.js"></script>
<script>

$("#DateCountdown").TimeCircles();
          $("#CountDownTimer").TimeCircles({ time: { Days: { show: false }, Hours: { show: false } }});
          $("#PageOpenTimer").TimeCircles();

              setInterval(function(){
             var remaining_second = $("#CountDownTimer").TimeCircles().getTime();
```

```
            if(remaining_second <= 0)
            {
              $("#CountDownTimer").TimeCircles().stop();
              alert('Your examination time is completed.\nYou are Redirecting to Result page.');
              window.location.assign("result_count.php");
            }
          }, 1000);


</script>


#Header Page
<?php
include '../controller/script.php';

$email = $_SESSION['email'];
$query1 = mysqli_query($conn,"SELECT * FROM admin_table WHERE email_address = '$email'");
$row  = mysqli_fetch_assoc($query1);
$fname = $row['firstname'];
$img =   $row['picture'];
?>


<head>
    <meta charset="utf-8" />
    <title>Dashboard | Smart CBT </title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta content="Premium Multipurpose Admin & Dashboard Template" name="description" />
    <meta content="Themesbrand" name="author" />
    <!-- App favicon -->
    <link rel="shortcut icon" href="../assets/images/favicon.ico">
    <link href="../assets/libs/dropzone/min/dropzone.min.css" rel="stylesheet" type="text/css" />

    <!-- jquery.vectormap css -->
    <link href="../assets/libs/admin-resources/jquery.vectormap/jquery-jvectormap-1.2.2.css" rel="stylesheet" type="text/css" />

    <!-- Bootstrap Css -->
    <link href="../assets/css/bootstrap.min.css" id="bootstrap-style" rel="stylesheet" type="text/css" />
    <!-- Icons Css -->
    <link href="../assets/css/icons.min.css" rel="stylesheet" type="text/css" />
    <!-- App Css-->
    <link href="../assets/css/app.min.css" id="app-style" rel="stylesheet" type="text/css" />
    <link href="../assets/libs/datatables.net-bs4/css/dataTables.bootstrap4.min.css" rel="stylesheet" type="text/css" />
     <link href="../assets/libs/datatables.net-buttons-bs4/css/buttons.bootstrap4.min.css" rel="stylesheet" type="text/css" />

     <link rel="../stylesheet" href="assets/style/bootstrap-datetimepicker.css" />
        <script src="../assets/style/bootstrap-datetimepicker.js"></script>

        <!-- DataTables -->
    <link href="assets/libs/datatables.net-bs4/css/dataTables.bootstrap4.min.css" rel="stylesheet" type="text/css" />
    <link href="assets/libs/datatables.net-buttons-bs4/css/buttons.bootstrap4.min.css" rel="stylesheet" type="text/css" />

</head>

<body data-layout="detached" data-topbar="colored">

    <div class="container-fluid">
       <!-- Begin page -->
       <div id="layout-wrapper">

          <header id="page-topbar">
             <div class="navbar-header">
                <div class="container-fluid">
                   <div class="float-right">

                      <div class="dropdown d-none d-lg-inline-block ml-1">
                         <button type="button" class="btn header-item noti-icon waves-effect" data-toggle="fullscreen">
                            <i class="mdi mdi-fullscreen"></i>
                         </button>
                      </div>

                      <?php
```

```
                $d = mysqli_query($conn,"SELECT * from alerts  ");
                $n = mysqli_num_rows($d);


                ?>

                <div class="dropdown d-inline-block">
                    <button type="button" class="btn header-item noti-icon waves-effect" id="page-header-notifications-dropdown"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        <i class="mdi mdi-bell-outline"></i>
                        <span class="badge badge-danger badge-pill"><?php echo $n;?></span>
                    </button>
                    <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right p-0"
aria-labelledby="page-header-notifications-dropdown">
                        <div class="p-3">
                            <div class="row align-items-center">
                                <div class="col">
                                    <h6 class="m-0"><?php echo $n .'  '.'Notification';?>  </h6>
                                </div>
                                <div class="col-auto">
                                    <a href="view_notification.php" class="small"> View All</a>
                                </div>
                            </div>
                        </div>
                        <div data-simplebar style="max-height: 230px;">


                         <?php

                         $d = mysqli_query($conn,"SELECT * from alerts ");
                         while($r = mysqli_fetch_assoc($d)) {

                                echo '
                                <a href="#" class="text-reset notification-item">
                                    <div class="media">
                                        <div class="avatar-xs mr-3">
                                            <span class="avatar-title bg-success rounded-circle font-size-16">
                                                <i class="bx bx-badge-check"></i>
                                            </span>
                                        </div>
                                        <div class="media-body">
                                    <h6 class="mt-0 mb-1">'.$r['exam'].'</h6>
                                    <div class="font-size-12 text-muted">
                                        <p class="mb-1">'.$r['offense'].'</p>
                                        <p class="mb-0"><i class="mdi mdi-clock-outline"></i>'.$r['time'].'</p>

                                </div>
                                </div>
                                  </div>
                            </a>

                                ';
                             }
                              ?>

                        </div>
                        <div class="p-2 border-top">
                            <a class="btn btn-sm btn-link font-size-14 btn-block text-center" href="view_notification.php">
                                <i class="mdi mdi-arrow-right-circle mr-1"></i> View More..
                            </a>
                        </div>

                    </div>
                </div>

                <div class="dropdown d-inline-block">
                    <button type="button" class="btn header-item waves-effect" id="page-header-user-dropdown" data-toggle="dropdown"
aria-haspopup="true" aria-expanded="false">
                        <img class="rounded-circle header-profile-user" src="<?php echo '../images_admin/'.$img; ?>" alt="Header Avatar">
                        <span class="d-none d-xl-inline-block ml-1"><?php echo $fname; ?></span>
                        <i class="mdi mdi-chevron-down d-none d-xl-inline-block"></i>
                    </button>
                    <div class="dropdown-menu dropdown-menu-right">
                        <!-- item-->
```

```
                        <a class="dropdown-item" href="change_password.php"><i class="bx bx-lock-open font-size-16 align-middle
mr-1"></i>Change Password</a>
                        <div class="dropdown-divider"></div>
                        <a class="dropdown-item text-danger" href="logout.php"><i class="bx bx-power-off font-size-16 align-middle mr-1
text-danger"></i> Logout</a>
                      </div>
                    </div>

                  </div>
                  <div>
                    <!-- LOGO -->
                    <div class="navbar-brand-box">
                      <span style="color: white;text-indent: 50px; font-size: 20px; link:white; text-size:40px; text-shadow: 2px 2px 5px red;"
>SMART CBT</span>  <a href="index.html" class="logo logo-dark">
                        <span class="logo-sm">
                          <img src="assets/images/logo-sm.png" alt="" height="20">
                        </span>
                        <span class="logo-lg">
                          <img src="assets/images/logo-sm.png" alt="" height="17">
                        </span>
                      </a>

                      <a href="index.html" class="logo logo-sm">
                        <span class="logo-sm">
                          <img src="assets/images/logo-sm.png" alt="" height="20">
                        </span>
                        <span class="logo-lg">
                          <img src="assets/images/logo-sm.png" alt="" height="19">
                        </span>
                      </a>
                    </div>

                  </div>

                </div>
              </div>
            </header>


#Footer Page
<!-- End Page-content -->

            <footer class="footer">
              <div class="container-fluid">
                <div class="row">
                  <div class="col-sm-6">
                    <script>document.write(new Date().getFullYear())</script> © Smart CBT.
                  </div>
                  <div class="col-sm-6">
                    <div class="text-sm-right d-none d-sm-block">
                      Design & Develop by BeautifulCode Inc
                    </div>
                  </div>
                </div>
              </div>
            </footer>
          </div>
          <!-- end main content-->

        </div>
        <!-- END layout-wrapper -->

      </div>
      <!-- end container-fluid -->



      <!-- JAVASCRIPT -->
      <script src="../assets/libs/jquery/jquery.min.js"></script>
      <script src="../assets/libs/bootstrap/js/bootstrap.bundle.min.js"></script>
      <script src="../assets/libs/metismenu/metisMenu.min.js"></script>
      <script src="../assets/libs/simplebar/simplebar.min.js"></script>
      <script src="../assets/libs/node-waves/waves.min.js"></script>
```

```
  <script src="../assets/libs/dropzone/min/dropzone.min.js"></script>

  <!-- apexcharts -->
  <script src="../assets/libs/apexcharts/apexcharts.min.js"></script>

  <!-- jquery.vectormap map -->
  <script src="../assets/libs/admin-resources/jquery.vectormap/jquery-jvectormap-1.2.2.min.js"></script>
  <script src="../assets/libs/admin-resources/jquery.vectormap/maps/jquery-jvectormap-us-merc-en.js"></script>

  <script src="../assets/js/pages/dashboard.init.js"></script>

  <script src="../assets/js/app.js"></script>

  <script src="../assets/libs/moment/min/moment.min.js"></script>
  <script src="../assets/libs/bootstrap-editable/js/index.js"></script>

  <!-- Init js-->
  <script src="../assets/js/pages/form-xeditable.init.js"></script>
    <!-- parsley plugin -->
    <script src="../assets/libs/parsleyjs/parsley.min.js"></script>

    <!-- validation init -->
    <script src="../assets/js/pages/form-validation.init.js"></script>
    <script src="../assets/js/lga.min.js"></script>

    <!-- Required datatable js -->
    <script src="../assets/libs/datatables.net/js/jquery.dataTables.min.js"></script>
    <script src="../assets/libs/datatables.net-bs4/js/dataTables.bootstrap4.min.js"></script>
    <!-- Buttons examples -->
    <script src="../assets/libs/datatables.net-buttons/js/dataTables.buttons.min.js"></script>
    <script src="../assets/libs/datatables.net-buttons-bs4/js/buttons.bootstrap4.min.js"></script>
    <script src="../assets/libs/jszip/jszip.min.js"></script>
    <script src="../assets/libs/pdfmake/build/pdfmake.min.js"></script>
    <script src="../assets/libs/pdfmake/build/vfs_fonts.js"></script>
    <script src="../assets/libs/datatables.net-buttons/js/buttons.html5.min.js"></script>
    <script src="../assets/libs/datatables.net-buttons/js/buttons.print.min.js"></script>
    <script src="../assets/libs/datatables.net-buttons/js/buttons.colVis.min.js"></script>
    <!-- Responsive examples -->
    <script src="../assets/libs/datatables.net-responsive/js/dataTables.responsive.min.js"></script>
    <script src="../assets/libs/datatables.net-responsive-bs4/js/responsive.bootstrap4.min.js"></script>

    <!-- Datatable init js -->
    <script src="../assets/js/pages/datatables.init.js"></script>

    <script src="vassets/js/pages/dashboard-2.init.js"></script>

</body>
```