# AN AUTOMATED BROADSHEET AND SECURED RESULT MANAGEMENT SYSTEM

**NWAOBI, CHUKWUEMEKA OLUWATUNMISE**

**16010301040**

**BEING A PROJECT SUBMITTED IN THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES IN PARTIAL FUFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE**

**MOUNTAIN TOP UNIVERSITY,**
**IBAFO, OGUN STATE,**
**NIGERIA.**

**NOVEMBER, 2020.**

# CERTIFICATION

This Project titled, **AN AUTOMATED BROADSHEET AND SECURED RESULT MANAGEMENT SYSTEM**, prepared and submitted by **NWAOBI CHUKWUEMEKA OLUWATUNMISE** in partial fulfilment of the requirements of the degree of **BACHELOR OF SCIENCE** (Computer Science), is hereby accepted

____________________________ (Signature and Date)

DR. F. A. KASALI

Supervisor

____________________________ (Signature and Date)

DR. I. O. AKINYEMI

Head of Department

**Accepted as partial fulfilment of the requirements for the degree of BACHELOR OF SCIENCE (Computer Science)**

________________________ **(Signature and Date)**

**Prof A.I. Akinwande**

**Dean, College of Basic and Applied Sciences**

# DEDICATION

This project is dedicated to Almighty God for helping me reach a successful completion of this work.

# ACKNOWLEDGEMENTS

I owe my profound gratitude to God Almighty who gave the strength, wisdom and courage, divine help and provision to me from the beginning to the completion of this work. I express gratitude to my supervisor, Dr. Kasali, F. A, for her teachings, guidance, counsel and motherly support in ensuring the successful completion of this research. God bless you Ma.

My heart-felt gratitude goes to Dr. I.O. Akinyemi, Head of Department, Computer Science and Mathematics, the Dean, College of Basic and Applied Sciences – Prof. Akinwande, A. I., and all other staff members of the department of Computer Science: Prof. Idowu, P. A., Dr. Oladejo, F, Dr. Adamu, O. B., Dr, Okunoye, O. B., Dr. Oyetunji, M. O. and other members of staff.

I acknowledge the constant support of my mentors who had contributed to my academic achievement. I pray God would continue to increase their knowledge. I will forever be grateful to my parents Mr. and Mrs Nwaobi, T. I., who sacrificed wealth and enjoyable moments of their lives for the sake of my success to ensure I complete this programme in time; and my siblings - Ugochukwu, and Chioma Nwaobi for their prayers and support.

I want to thank all my Covenant University colleague and friends for their prayers and support, and help in one way or the other. God bless them all greatly.

I would not forget to remember all the students in the Department of Computer Science and Mathematics, for making my stay a worthwhile one, I say God bless you all richly.

## TABLE OF CONTENTS

# ABSTRACT

This project is based on the Design and Implementation of an Automated Broadsheet and A Secured Result Management System for use at the Mountain Top University. This project is based on enhancing the current system with more security technologies. With the help of the internet, computers and other mobile devices are used to manage result management and help the administrator effectively access these records.

This project aims to create an efficient and reliable result management system that can process students result with ease. In order to achieve its aim and objectives, a database was created, and design steps were taken using the iterative and incremental model.

This project work was built on Laravel framework with other frontend and backend technologies using Visual Studio Community 2019 as the Integrated Development Environment

The Automated Broadsheet and Secured Result Management system created is useful in helping administrators and lecturers monitor and manage their respective operations properly.

The Automated Broadsheet and Secured Result Management system is a faster and more efficient way of keeping students records and monitor them.

# LIST OF TABLES

## LIST OF FIGURES

Figure Page

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background to the Study

This research focuses on creating an automated broadsheet and a secured result management system using a biometric system, database, forms and representations. This is a computerized examinations result management system for tertiary student's examination records. The main method of students' academic result processing is found to be tedious, especially when carried out for a large number students, this makes the entire process time-consuming and error prone. An automated broadsheet system presents a single platform that is used to manage the processing of all academic records for any given institution. New Grading System, was established by the National University Commission (NUC) for Nigerian Universities**.**

| Marked Range | Letter Grade | Grade Point |
|---|---|---|
| 70%-100% | **A** | **5** |
| 60%-69% | **B** | **4** |
| 50%-69% | **C** | **3** |
| 45%-49% | **D** | **2** |
| 0-44% | **F** | **0** |

**Table 1.1: Grading System for University Courses in Nigeria**

| **CGPA** | **CLASS OF DIPLOMA** |
|---|---|
| 4.50 – 5.00 | First Class |
| 3.5 – 4.49 | Second Class Upper |
| 2.40 – 3.49 | Second Class Lower |
| 1.50 – 2.39 | Third Class |
| 1.0  - 1.49 | Pass |

**Table 1.2:  Cumulative Grade Point Average**

The courses offered in for a B.SC program are allocated a number of credit units which vary from one course to another, because the courses vary in their needs and scope. Hence some are allocated greater credit units than others. The grade-points obtained by a student in any course are determined by multiplying the value of the grade (numeric grade) by the credit units of the course. The total grade-points are obtained by summing up the grade-points of all the courses offered. The basic formula for calculating GPA is to divide the total points earned in a program by the total number of credits attempted. The resulting figure is the GPA for that program. Thus: reference (NUC,

$$G = {}^{Total\ points\ for\ courses\ offered}\Big/_{\sum \quad credit\ unit\ per\ semester\ offered}$$

$$Gc = {}^{\sum Total\ points\ obtained}\Big/_{\sum_n Credit\ units\ from}$$

Where;

n=Numbers of Year

$G = GPA$

$G_c = CGPA$

## 1.2 Statement of the Problem

Many challenges are faced with the manual method of keeping records. There is no structure that compels students to register all outstanding courses before proceeding to the recent ones. The challenges faced with this manual system include time wastage in searching for registers, inadequate use of statistical data, error prone calculations, and redundancy of information. This project is aimed at devising a system that will eradicate these above problems and improve result management. This computerization process is believed to be capable of not only solving these problems but many more to be encountered.

## 1.3 Aim and Objectives of the Study

The aim of this study is to create a Result Management System (RMS) and broadsheet generator that can do all the manual work with ease. It is designed to achieve the following objectives:

i. Design a model for an efficient, reliable and better secured RMS

ii. Implement the model

## 1.4 Scope of the Study

The scope of the project covers the development of a web-based database application for use at the Mountain Top University ICT department to replace their old result management system. The requirements include designing a user interface for the application and providing biometrics system for authorized access to the lecturers; register a new student and view a list of already registered students; to keep records of students in the department; view students registered on the database; passed courses, outstanding courses, required units for graduation etc. It also covers writing the background programming to ensure that the interface works with the database through the underlying codes to perform the required actions. It also involves the testing, improvement and optimization of the application.

### 1.5 Significance of the Study

The researcher during the course of this investigation found out that the existing RMS used in academic institutions can be improved on by adding more security through the use of biometrics. Result computation would be easier and convenient for the exam officers as all result information will be kept and generated from the system.

The new system would enhance the result processing performance as it will reduce delay in computing student's result coupled with the better security.

The significance of this study is to improve collating of students result, generation of result broadsheet electronically which would inadvertently speed up the release of results to students in academic institutions especially with the COVID19 pandemic which have necessitated most institutions to embrace the blended learning approach.

### 1.6 Definition of Terms

**Automated Broadsheet System:** An automated Broadsheet generator is a software designed to manage all areas of a result processing systems.

**Record Management:** Records Management refers to an on-going process of managing the records in a media neutral basis in accordance with approved policies, procedures and schedules

**Repository:** A central location in which data is stored and managed

**Database:** A collection of organized information in a regular structure, usually but not necessarily in a machine-readable format accessible by a computer.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.0 Introduction

This part gives a survey of writing on the components affecting the execution of A secured Automated Broadsheet and result management system. The introduction of this section starts with the conceptual review, theoretical review, review of related literature and the summary of the literature reviewed.

A Result Management System handles all students results scores, passed courses, exceptional courses and required units for the department for graduating. Several result management systems used to manage student data have been referred to in various ways: Student Information Systems (SIS), Student Management Information Systems (SMIS), Student Data Systems (SDS), Student Data Warehouse (SDW), Student Academic Information Systems (SAIS), or Student Information Management Systems (SIMS), (Ngoma, 2009). A result management system (RMS) is only an aspect of a complete SIS package.

The U.S. Branch of Training (2008) sees result Management system as incorporating "equipment and programming that give various capacities to clients, for example, putting away current and chronicled information, quickly arranging and breaking down information, and creating introduction configurations or detailing interfaces

The educational sector is an area of economic interest in several countries; therefore, there have been lots of efforts in the use of automated result management. All things considered, there is prove proposing that these security measure have not been embraced as anticipated, and although there are some proposals to support their adoption, the proposed support is not by means of information and communication technology which can provide automatic tools of support.

The aim of this work is to plan and execute a university's RMS, this system will have the capacity of putting away current and recorded information, organize and analyze this data as, provide an adequate security measure before login to those who have access and the software will have the ability to be integrated with other SIS packages if the need emerges.

## 2.1 Conceptual Review

### 2.1.1 Result

The word result is derived from the Latin word "consequuntur", which relates to a final score, mark, or placing in a sporting event or examination. A result is the final grade of students in tertiary institution after series of tests and examination has been taken in their course of study then finally graded. It is also defined as the standard Schedule of current assessment grades and administrative codes contains all letter grades (and their scores)

### 2.1.2 Result Management

Result Management is a wide-ranging term that can be defined from various aspects. It is primarily about managing all aspects of results coordinating all elements of an institution. This can range from Exam officer to record keeping to inventory of grades and CGPA of students. All sections of the educational system require information processing. With the use of computers for information processing, instant access to the academic and administrative information will be possible.

Result management is a process by which students' grades will be examined on the courses they registered for and the result managed and properly stored.

It is generally known that the imbalance between man power availability and the load of work to be done in processing examination results, leads to the delay in the declaration of examination results.

The effective measure, which can improve the efficiency of the examination result processing is therefore the introduction of computerization

### 2.1.3 Result Management System

Result Management System is an advanced opportunity for the school, college, university and coaching center institutions searching for a secure, simple and alternative solution to the conventional paper-based exam results evaluation, reporting and distributions. The system comes with certain advantage and disadvantage.

The Result Management System has unique templates by providing the administration with a secure database system for storing, evaluating and publishing the test scores and grades of students online.

The database likewise allows the storage students of student's scores and data then generate the exam results on the web at whatever point necessary.

The software solves and eradicating the problems in traditional result and examination management with automation system. One of the main purposes of the system is to give the test results to the students in the as fundamental and exact way as possible. The Exam Officer and lecturers can likewise modify and assess marks for the students whenever needed.

Result management system (RMS) handles all Lecturer, students, grades, course code and course title information. An RMS system helps manage result processing of the university.

Ajay and Abhishek (2012), a good database does not permit anomalies and uses an orderly fashion to save relevant information for data integrity. Hence, for accuracy and ease of retrieval, tables have to be normalized.

They also developed their solution using PHP embedded in HTML and linked to MySQL database with a PhpMyAdmin that can run on both WAMP and LAMP. PHP which is a server-side scripting language was preferred because of the ease of use on diverse platforms with minimal change to the script; and for its compilation speed and efficiency. Operations of a result management system to function smoothly, such as:

i) It makes the workload lighter

ii) It makes it more secured

iii) It enables accurate processing of result information

iv) It allows for better competitiveness

## 2.2 Theoretical Review

Okonigene et al. (2008) stated that, the use of computers for information processing, the following are possible: instant access to students' personal and course information, instant student information updating, automatic computation of the Grade Point Average (GPA), generation of the graduating students list, monitoring of failed courses, keeping an up-to-date record of the entire student body in Mountain Top University, storing of different course information such as course code, course description, course unit, and scores for the purpose of Grade Point Average computation, and producing user friendly data entry screens for ease of use. With the suitable software in place, the system would simply capture raw scores as entered by individual lecturers for various students in the different courses, and then process accordingly.

Vecchioli (1999) also stated that organizing and managing student records into a cohesive and efficient system might seem like an impossible task. This study was carried out to verify the manual process involved in generating student's examination result then adding security measures to the software so those without permissions won't have access to it and to seek way of automating the system for effective operation.

Student Examination Result Processing System (2012) explains that the effort put in place in the process of registration of students and computation of their examination results is enormous. Therefore, the need to develop a computerized process that will effectively and efficiently capture all the important data associated with the registration and examination result processing within the school.

Barrett (1999) also stated that in an effort to efficiently document and maintain accountability data, schools are relying more on technology in the form of Student Management Information System (SIMS). This result management system is designed to efficiently handle processes like inputting scores, storing results, automatically calculating grade points, and interpreting the student's overall result.

The manual process has now reached a level where it is difficult for the available man power to cope with the magnitude of examination work, in a given time.

An effective measure, which can improve the efficiency of the examination result processing, is therefore the introduction of computerization, especially with the use of examinations result processing software. Computerized relational database systems like MySQL and a Scripting Language.

## 2.3 Review of Related Works

Chidinma (2015) noticed the problems involved in the manual method of result management system at University of Nigeria, NSUKKA which were unreliable backup of file, difficulty in accessing information which would result in waste of time and inaccuracy of reports. She created a result management system capable of improving time of results collation, reducing transcription errors, reducing duplication of information entries, optimizing reporting time, reducing pilfering chances and maintaining records of each student. The system was built on Java in which a waterfall model was followed for the development of the system.

Though the system met the minimum expectations that were initially set for it, but it also had some limitations which were the system could not be used in all areas of the university, the system was not user friendly and the system was vulnerable to attacks.

Lorgat (2018) focused his research on implementing based on the software architecture standard, MVC, which describes its three layers. The information flow of this project using the MVC standard highlights in the View layer, that is, the client section where information or resources are requested from the system. The Controller layer is responsible for receiving the requests and then processing and directing them to the Model layer in charge of satisfying the request by retrieving the information from the database. Then it passes the information obtained, to the Controller which delivers the response to the View and finally displays the information to the client, through the browser.

Ekanem and Ozuomba (2017) implemented a web-based result management system for The University of Uyo. Their aim was to increase the efficiency and interactivity in any area of specialization in the result processing. The spiral software development model was adopted and used in this system.

The system was developed using PHP programming language and Visual Studio 2018 as its development environment with MySQL as its database. The system is able to perform the following functions:

i) The system can authenticate the users of the system.

ii) Only the administrator can make changes to the database

iii) The system was able to generate test reports, Student continuous assessment details and Exam scores

Due to inadequate facilities used, the system was unable to provide a mobile friendly user interface.

Result Management System is one of the several products created by the software developers at Codecanyon. It is a complete software for tertiary institutions. It supports laptop, smartphone, desktop and tablet devices. It integrates and facilitates seven types of user area of a university, namely Administrator, Lecturer, and Student. The software also includes a security feature claimed to be invulnerable to threats such as SQL-injection, XSS attacks and CSRF (Codecanyon, 2018). Improvements made to Result Management System by the proposed system include reduced data redundancy possibilities and cost efficiency. Due to limited time, the design of the system was not responsive on mobile devices.

Morningsunit result Management System has the following features: Student and teacher information, subject and department information, signup & sign in, online notification alert, released results alert, Easy password changing facility. Student and teacher information: this contains information about the student and lecturers based on their profile input, Subject and department information.

This contains information on each course taken by the students and information about them. Online Notification alert: this alerts the admin when a user login into the system for security purpose. Released result alert: this sends a notification to students by mail so they know when their results are released. Easy password changing facility: this helps in changing of passwords easily without any prior stress

However, the system had a nice user interface design implemented but it had a slow page load speed.

# CHAPTER THREE

# METHODOLOGY

## 3.0 Introduction

This chapter provides a methodology of the automated Broadsheet and result management system to be built. The presentation of this chapter begins with the flowchart, use case diagrams, design details, software development life cycle and the methods of data collection.

The design of a secured result management system, is to improve the quality of services rendered in the University. With the improved changes of information and technology, the process by which automated management systems are made is changing dramatically. Using ICT in Tertiary institutions has led to the improvement of accurate result management. Coupled with the rapid changes of ICT evolution in the society, the tertiary institution should be along with the changes of modern society too. Tertiary intuitions have benefitted from the use of Information Technology (I.T).

## 3.1 Flowchart of the Result Management System

The flowchart illustrates the flow of control in program modules. It is a visual or symbolic representation of a process. The flowchart does not mention anything about how data flows through the system.

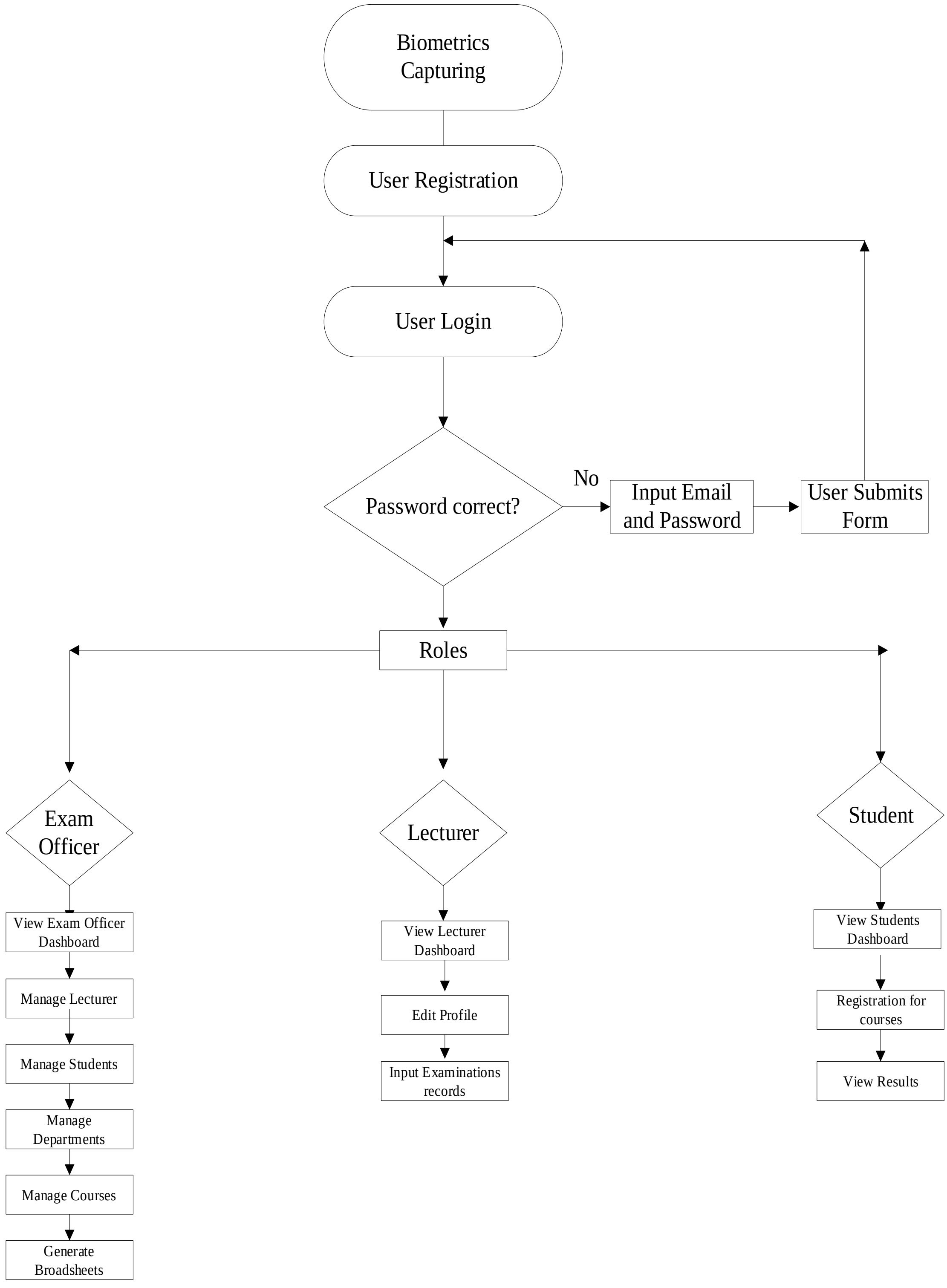


**Figure 3.1 Flowchart of the Secured Result Management System**

## 3.2 Use Case Diagram

A use case diagram is a graphical representation of the relationship between the elements of a system. A use case is also a methodology used in system analysis to define, clarify, and organize system requirements (Tech Target, 2020). The following are the different use case diagrams for the system:

### 3.2.1 Exam Officer (Admin) Use Case

The Exam officer shall be able to login into the system, interact with the dashboard and view what the system has to offer, Admin shall be able to manage Lecturers, manage courses, process results, manage levels of students, manage students' profile and also generate broadsheets of each course in different departments

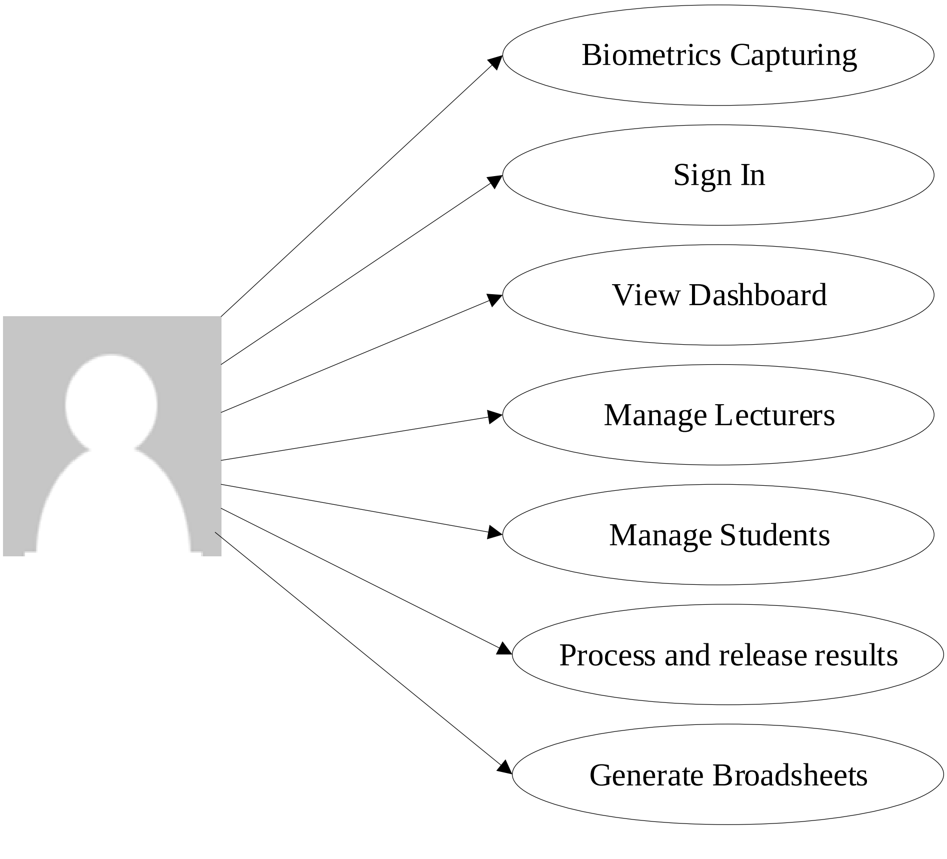


**Figure 3.2 Exam Officer (Admin) Use Case**

### 3.2.2 Lecturer Use Case

The Lecturer shall be able to login into the system, interact with the dashboard and view what the system has to offer, Lecturer shall be able view the dashboard, edit profile and submit students' scores/records on each course taken and make the system generate the student's individual GPA

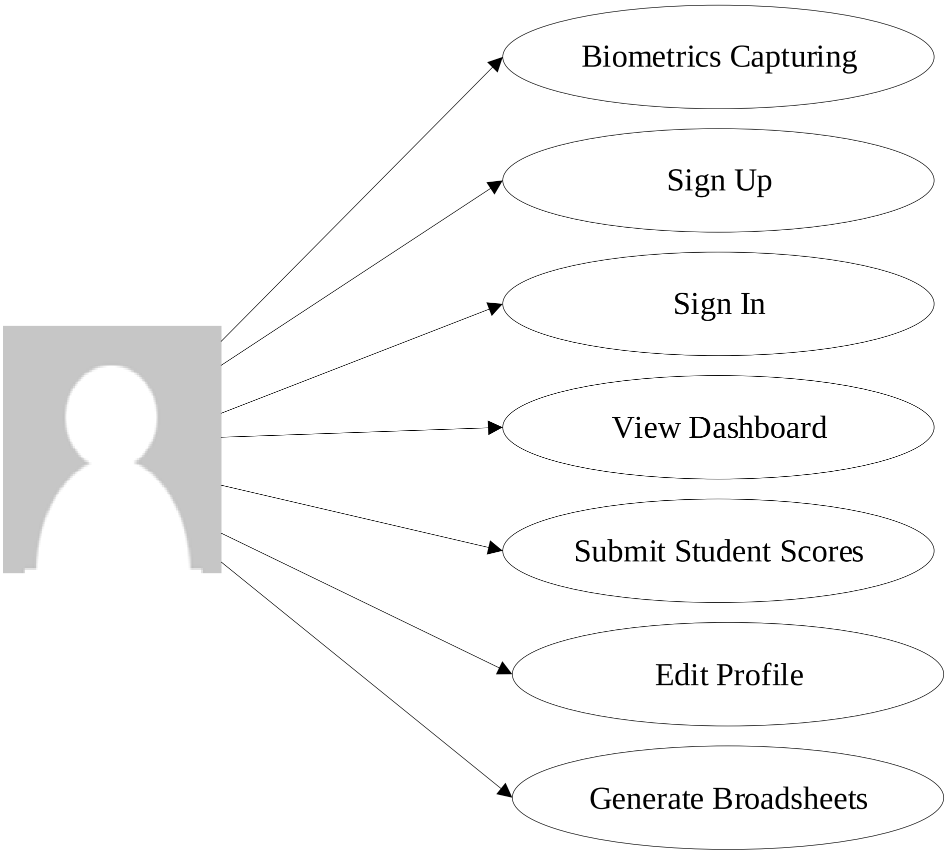


**Figure 3.3 Lecturer Use Case**

### 3.2.3 Student Use Case

The students shall be able to login into the system with their individual Matric Number and self-assigned password, interact with the dashboard and view what the system has to offer, Students shall be able to register for the courses to be taken for the semester, edit their profile and check individual results

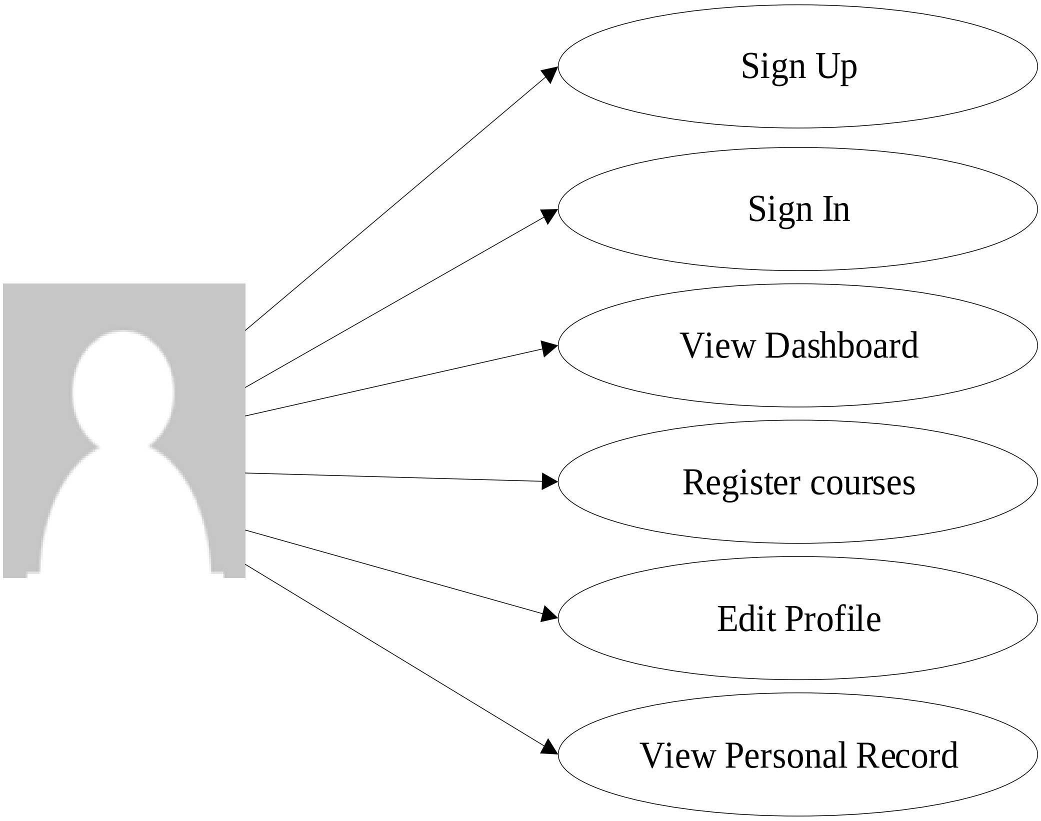


**Figure 3.4 Student Use Case**

## 3.3 Design Details

The automated Broadsheet and result management system was developed on Laravel framework. Laravel is a Model-View-Controller (MVC) PHP framework used to developed web applications. The MVC is an architectural pattern that separates an application into three main logical components: The Model (data), the View (user interface), and the Controller (application logic). This pattern helps to achieve separation of concerns. The Model part is responsible for managing the data of the application. The Controller is responsible for controlling the application logic and acts as the coordinator between the View and the Model, and the View is the component involved with the application’s user interface. Using the MVC pattern for websites, requests are routed to a Controller which is responsible for working with the Model to perform actions and/or retrieve data. The Controller chooses the View to display, and provides it with the Model. The View renders the final page, based on the data in the Model. Laravel has a very rich set of features which will boost the speed of web development. (Tutorialspoint, 2019).

## 3.4 Software Development Life Cycle

The iterative and incremental model was adopted and used for the development of the system. In this model, the iterative process begins with a simple implementation of a small set of software requirements and iteratively enhances the evolving versions until the entire system is implemented and ready for deployment. An iterative life cycle model does not attempt to begin with a full requirements specification. Instead, development begins with only part of the software specified and implemented, which is then reviewed to identify additional requirements.

This process is then repeated and at the end of each iteration model, a new version of the software is produced. The following diagram is a representation of the iterative and incremental model. (Tutorialspoint, 2019).

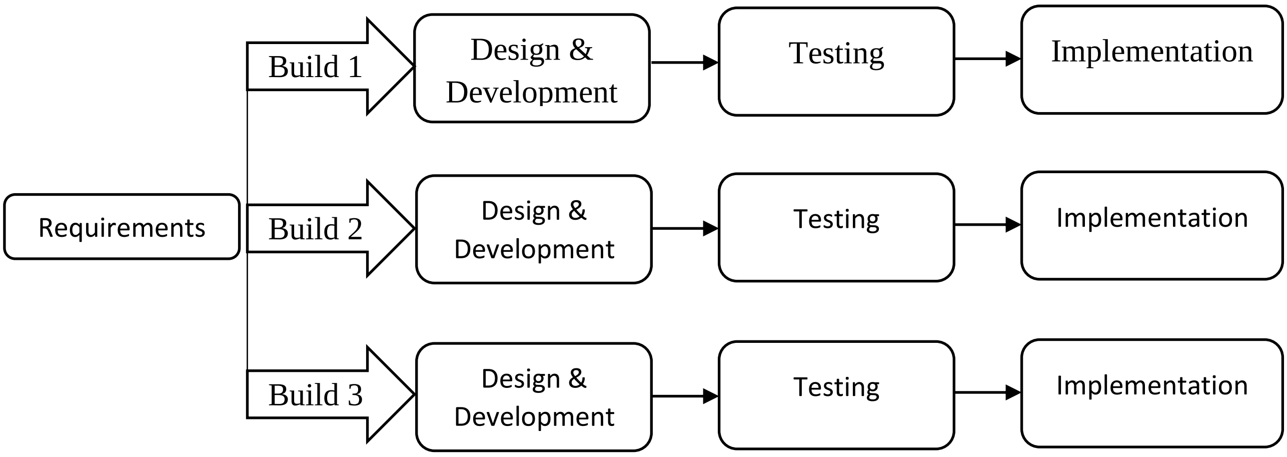


**Figure 3.5 Iterative and Incremental Model**

The advantages of this model are as follows:

i. Some working functionality can be developed quickly and early in the life cycle.

ii. Results are obtained early and periodically.

iii. Parallel development can be planned.

iv. Progress can be measured

v. It is easy to test and debug during smaller iteration

vi. The operational product is delivered with each increment

This model has the following disadvantages:

i. It may require more resources.
ii. It is not suitable for smaller projects.
iii. Although cost of change is lesser, but it is not very suitable for changing requirements.
iv. Highly skilled resources are required for risk analysis.
v. More management attention is required.
vi. Projects progress is highly dependent upon the risk analysis phase.

## 3.5 Method of Data Collection

The method of data collection used for this project work is from secondary sources. The data would be collected from the Information and Communication Technology (ICT) department of Mountain Top University

# CHAPTER FOUR

# IMPLEMENTATION AND RESULT

## 4.0 Introduction and documentation

This chapter demonstrates the information of implementing an automated broadsheet and result management system. Designing and implementing an automated broadsheet and result management system considers some aspect which attempt to provide alternatives to the issues recognized and indicated in result processing systems. It describes the tools used in developing and implementing the system. These tools assisted in system design and development of the system's primary idea and functionality to accomplish its defined mission.

It describes a variety of functionalities step beneath every module with their outputs. The entire notion is a system, which are set of things working together as components of a mechanism or an interconnecting system, which are set of matters working collectively as parts of mechanisms or an interconnecting system, it is a set of interacting or interdependent issue forming a set of elements to other elements.

## 4.1 Implementation

The system was built using Laravel MVC Framework. The implementation was divided into three main parts which are frontend, backend and database. The frontend was built using HTML, CSS, JavaScript and jQuery. The frontend files were stored in the views folder of the MVC framework. The backend was built using PHP with Laravel Framework.

The backend codes were kept in the Controller folder. The MySQL database was used for the implementation of the system in which the database codes were stored in the Model folder of the framework. Git was used to track changes made to the code.

## 4.2 How the Result Management System works

The application works based on the functional and non-functional requirements. This application includes the major phases of software development lifecycle.

### 4.2.1 Functional Requirements

This section describes the system's anticipated features. To execute a number of high-level tasks. For the planned use of this application, the following specifications are expected.

i. **Biometric Scanning Device**: The device scans the registered Exam officer or Lecturer finger templates by saving it to the database, then tries to match it with the record saved on the database whenever an authorized person wants to login into the system.

ii. **Registration**: All users of the system can successfully register on the system by filling in their details.

iii. **User Management**: Modify rights, creating user ids, adding/deleting existing users.

iv. **Result Management**: Admin or the lecturers can Modify student scores

v. **Uploading of score sheets**: Lecturers can upload the results of each departments through an excel sheet while the system sorts the scores out for each student and grades them on their GPA

vi. **Downloading Files**: Admin can download the list of all users in the system. students can also download all their processed results.

vii. **Backing Up Files:** The system automatically backs up files upon every change made on any student result, and Git is implemented to track several changes.

### 4.2.2 Non-Functional Requirements

This deals with the characteristics of the system, which cannot be expressed as functions.

i. **Security:** Access permissions for users are only granted when the email matches with password.

ii. **Reliability:** The database update process must rollback all related updates when any update process fails.

iii. **Performance:** The webpage load time is not more than 9 seconds for users that access the system

## 4.3 System requirements

For the efficient use of the system, certain hardware components and software components must be present on the system. The system requirement is made up of the software and hardware parts that makes the system to be developed effectively.

### 4.3.1 Software Requirements

One of the key elements in building a system is the section of compatible software. The following software are recommended for the successful implementation of the system.

| | |
|---|---|
| Front-end technologies: ` | HTML, CSS, Bootstrap, JavaScript, jQuery, |
| Backend technologies: | PHP, Laravel Framework 7, Node Js and Vue Js |
| Database Management System: | phpMyAdmin, MySQL |
| Version Control Tool: | Git |

| | |
|---|---|
| Local Server: | XAMPP |
| IDE: | Visual Studio Code 2019 |
| Web Browser: | Google Chrome, Mozilla Firefox, Safari, IE |

### 4.3.2 Hardware Requirements

Hardware configuration is an important factor to put in mind when developing a system. Insufficient random-access memory may affect the speed and efficiency of the entire system. The processor should be powerful to handle the entire operations. The hard disk should also have sufficient capacity to store the file and application.

| | |
|---|---|
| Processor: | Core i3 (Minimum) |
| Processor speed: | 2.5GHz (Minimum) |
| RAM: | 4GB (Minimum) |
| Hard disk: | 500GB (Minimum) |
| Monitor Display: | LED |
| Mouse: | Touchpad with multi-touch gesture support, USB or PS/2 |
| Raspberry Pi: | Biometric Scan |

## 4.4 Screenshot of the Pages Implemented

The screenshots of the implemented pages show the different views of the users depending on their roles with a brief description of what it entails. It consists of the Authentication Pages, Exam Officer (Admin) Dashboard, Lecturer Dashboard and students Dashboard

### 4.4.1 Authentication Pages

The authentication pages consist of both the signup and login page

#### i) Biometric Scan Page

This is the first page that appears when the URL of the RMS is typed in any browser. The biometric interface is used to register users' fingerprint templates before giving them access to their privileges. When the Biometric system has scanned the users fingerprint template, the system redirects the users to their respective dashboard based on the roles selected or returns the user to a 404 page if the fingerprint scan doesn't recognize the users finger template.

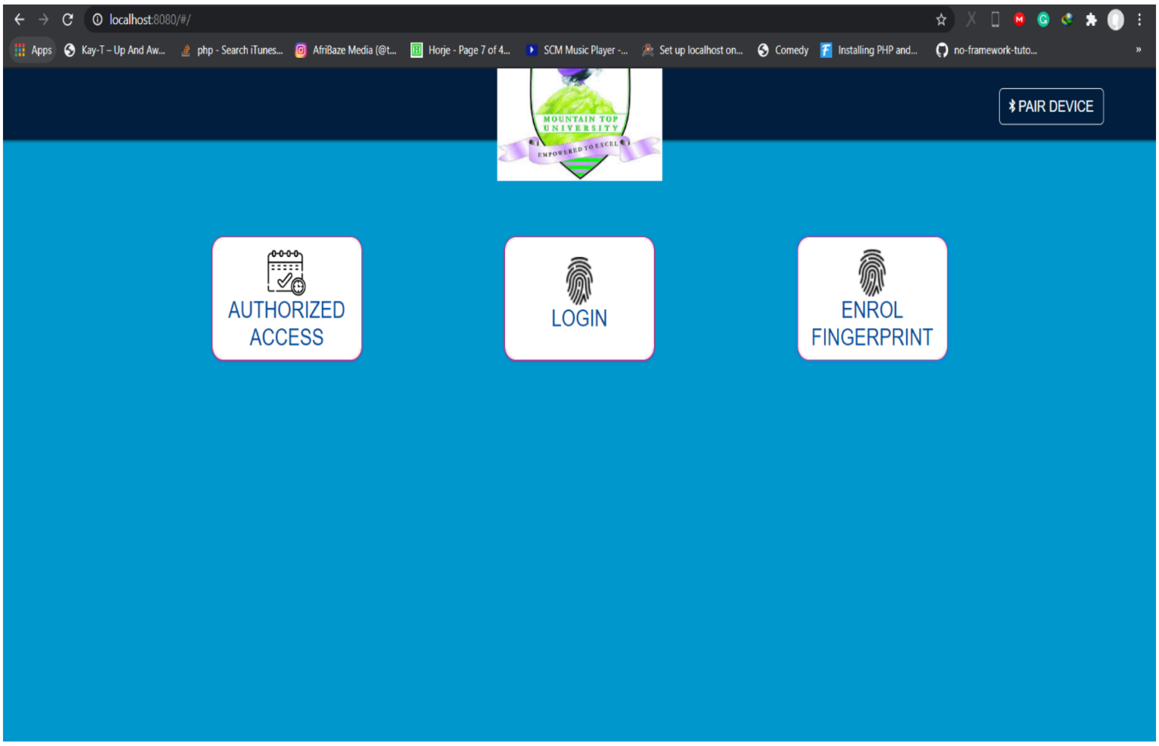
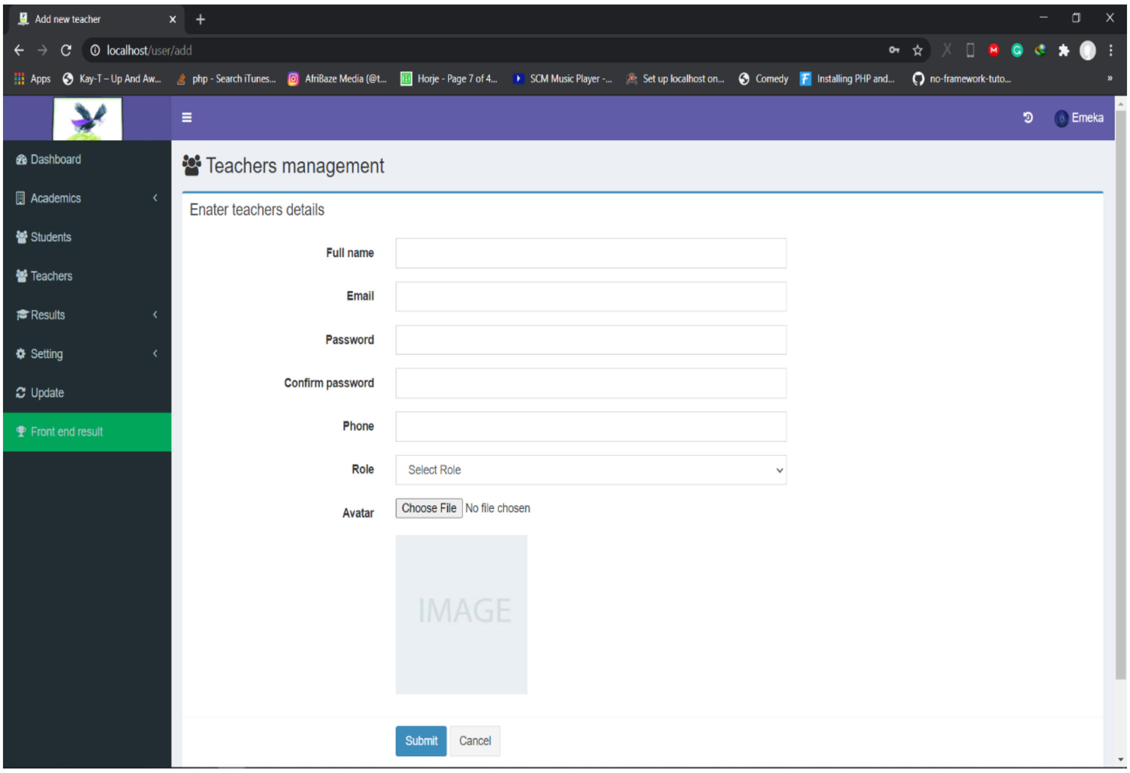


**Figure 4.1 Biometric Scan Page (Admin)**

### ii) Signup Page

This is the first page that appears when the URL of the RMS is typed in any browser. The signup form is used to register users’ credentials before giving them access to their privileges. The signup form contains a field for the name, email, password and role. When the signup form is submitted, the system redirects the users to their respective dashboard based on the roles selected.



Teachers management

Enater teachers details

Full name

Email

Password

Confirm password

Phone

Role: Select Role

Avatar: Choose File No file chosen

**Figure 4.2 Sign Up Page (Admin)**

### iii) Login Page

The login form is used to authenticate users’ credentials before giving them access to their privileges. The login form contains a field for the email and another for the password. When the login form is submitted, the code checks that the credentials are authentic, given the user access to the restricted page. If a user is not authenticated correctly, they will not be able to proceed past the login form.

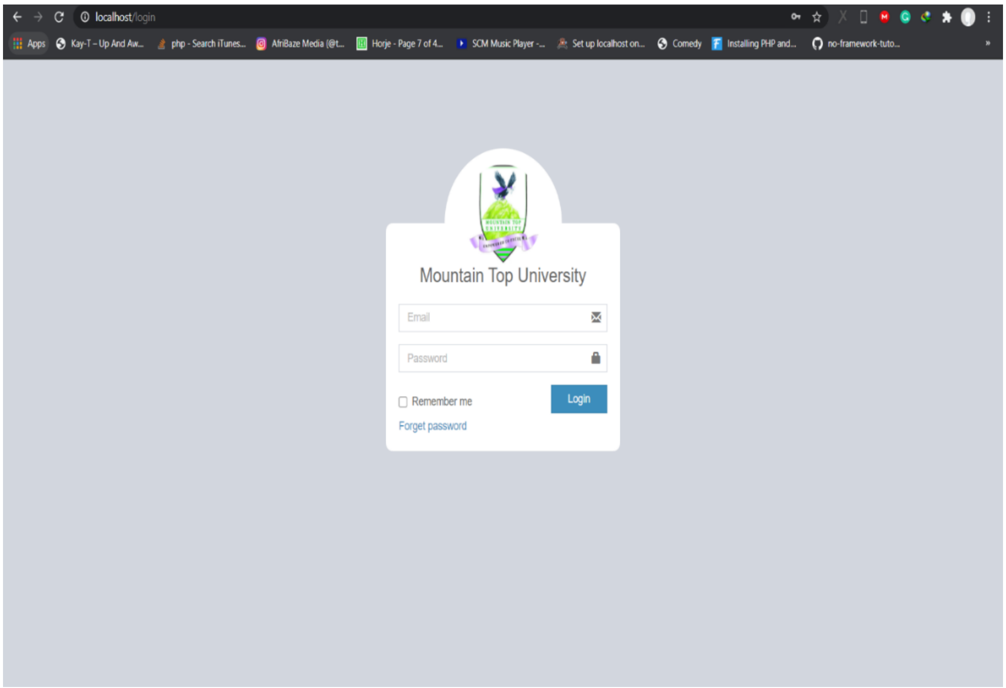


**Figure 4.3 Login Page (Admin)**

### 4.4.2 Exam Officer (Admin) Dashboard

The Exam officer dashboard is tasked with the management of all users of the system. The Exam Officer can add users, edit users' profile as well as delete users from the system.

### i) Home

In this page, the Exam Officer can see the total number of users of the system.

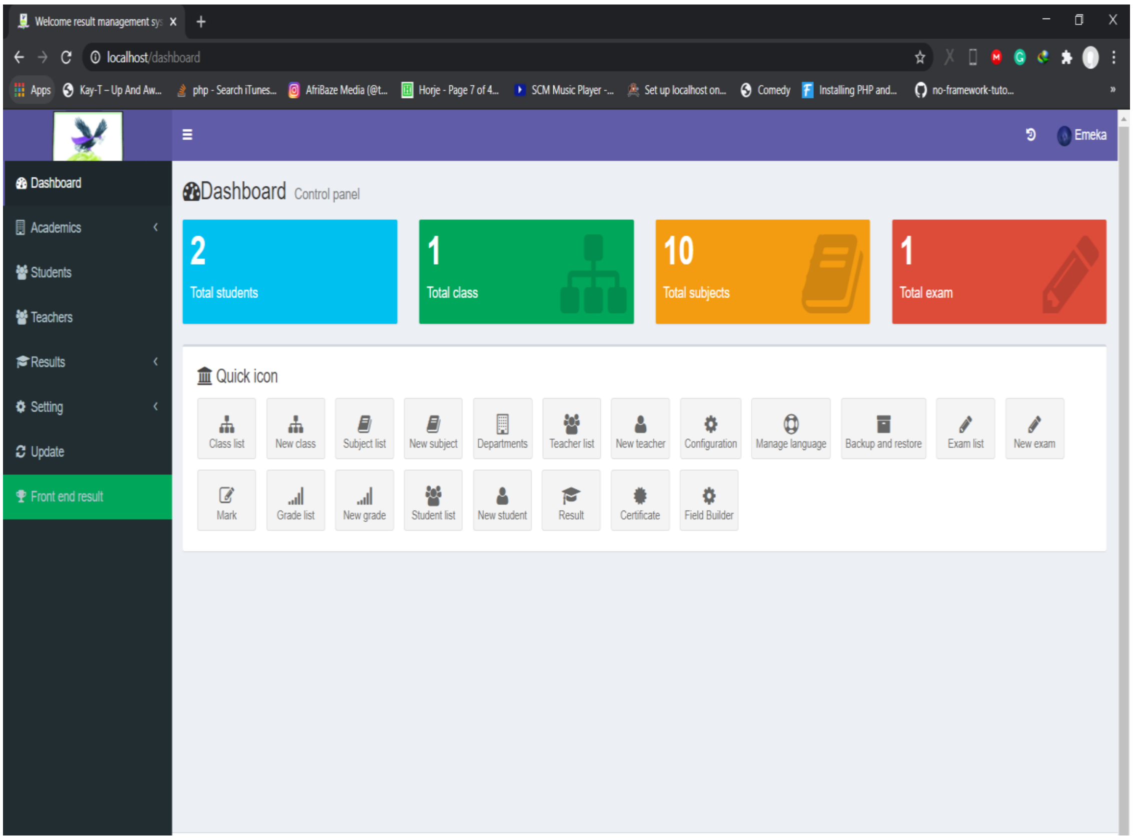
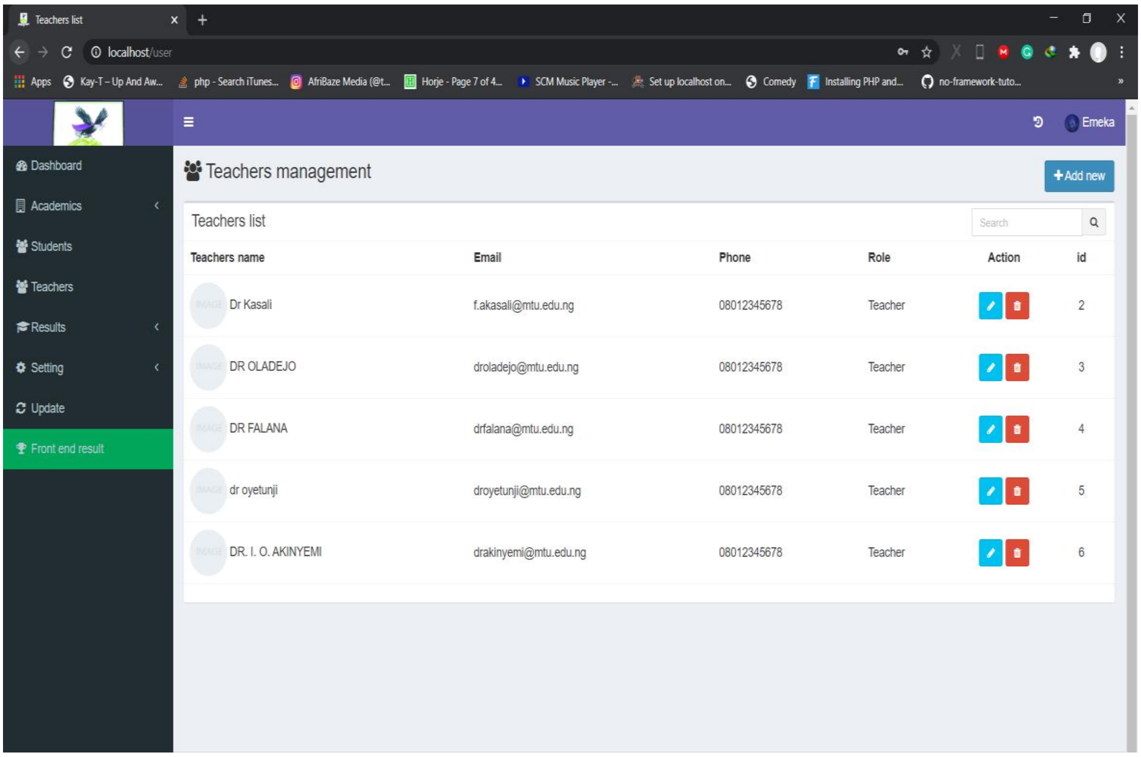


**Figure 4.4 Exam Officer Dashboard (Admin)**

### ii) Lecturers

In this interface, the Exam Officer is tasked with managing the lecturers in the university. The Exam officer can add lecturers, edit lecturers' profile as well as delete lecturers from the system.



Teachers management

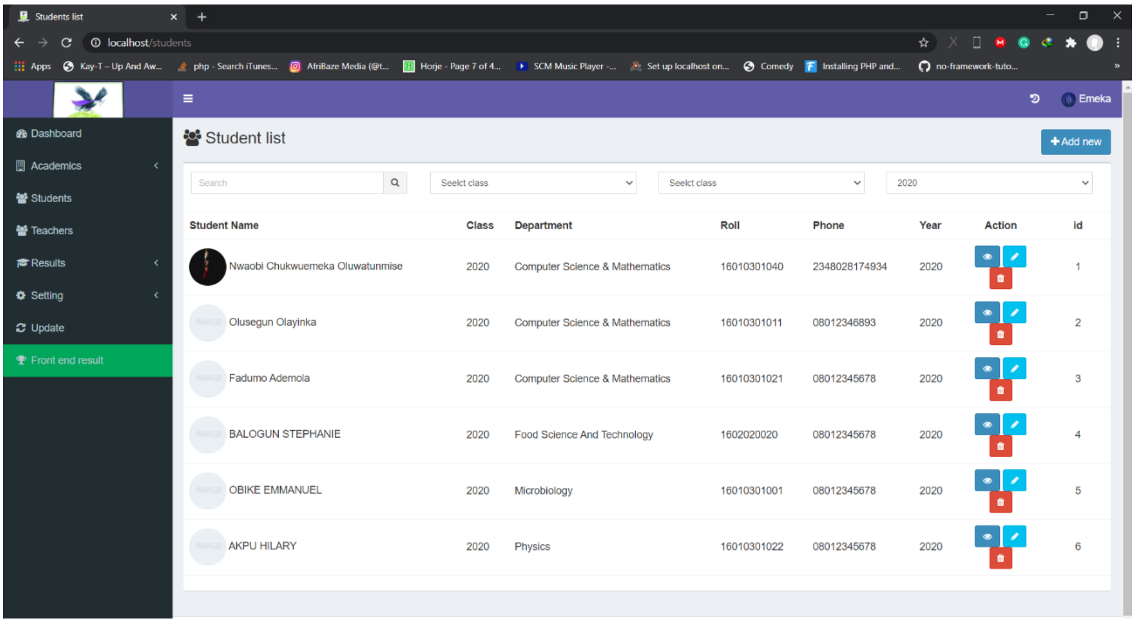
Teachers list

| Teachers name | Email | Phone | Role | Action | id |
|---|---|---|---|---|---|
| Dr Kasali | f.akasali@mtu.edu.ng | 08012345678 | Teacher | | 2 |
| DR OLADEJO | droladejo@mtu.edu.ng | 08012345678 | Teacher | | 3 |
| DR FALANA | drfalana@mtu.edu.ng | 08012345678 | Teacher | | 4 |
| dr oyetunji | droyetunji@mtu.edu.ng | 08012345678 | Teacher | | 5 |
| DR. I. O. AKINYEMI | drakinyemi@mtu.edu.ng | 08012345678 | Teacher | | 6 |

**Figure 4.5 Lecturer list (Admin)**

### iii) Students

In this interface, the Exam Officer is tasked with managing the students in the university. The Exam officer can add students, edit students' profile as well as delete students from the system.



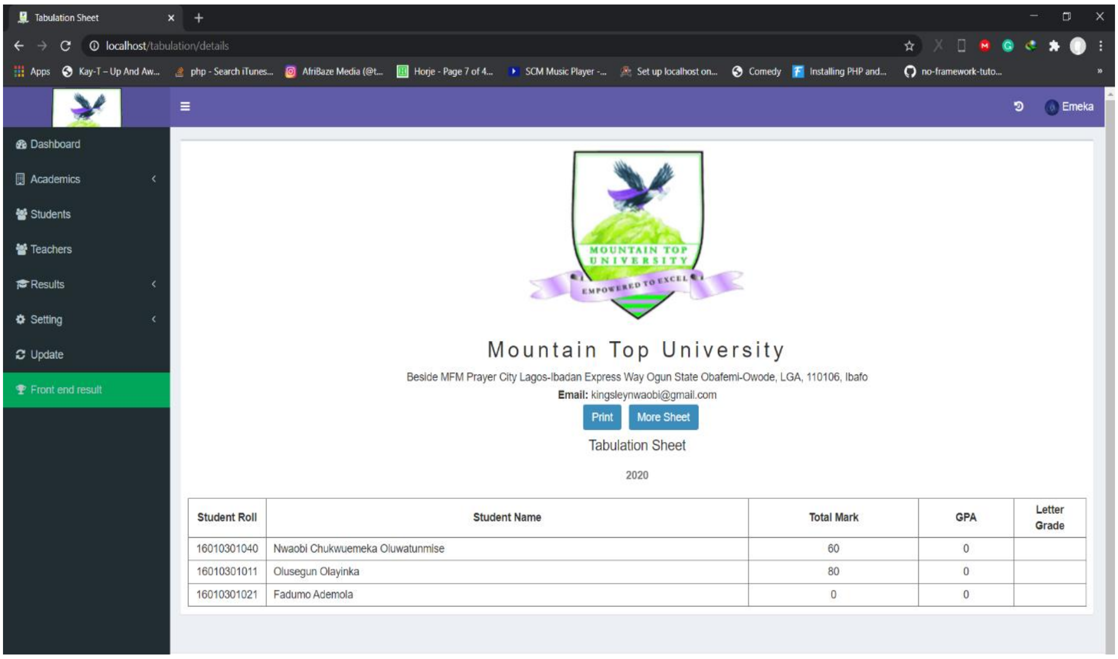
Student list

| Student Name | Class | Department | Roll | Phone | Year | Action | id |
|---|---|---|---|---|---|---|---|
| Nwaobi Chukwuemeka Oluwatunmise | 2020 | Computer Science & Mathematics | 16010301040 | 2348028174934 | 2020 | | 1 |
| Olusegun Olayinka | 2020 | Computer Science & Mathematics | 16010301011 | 08012346893 | 2020 | | 2 |
| Fadumo Ademola | 2020 | Computer Science & Mathematics | 16010301021 | 08012345678 | 2020 | | 3 |
| BALOGUN STEPHANIE | 2020 | Food Science And Technology | 1602020020 | 08012345678 | 2020 | | 4 |
| OBIKE EMMANUEL | 2020 | Microbiology | 16010301001 | 08012345678 | 2020 | | 5 |
| AKPU HILARY | 2020 | Physics | 16010301022 | 08012345678 | 2020 | | 6 |

**Figure 4.6 Students Page (Admin)**

### iv) Generate Broadsheets

In this interface, the Exam Officer is tasked with generating result broadsheets of various departments and specific year for each exam in the university. The Exam officer can add and delete broadsheets from the system.



Mountain Top University

Beside MFM Prayer City Lagos-Ibadan Express Way Ogun State Obafemi-Owode, LGA, 110106, Ibafo

**Email:** kingsleynwaobi@gmail.com

Tabulation Sheet

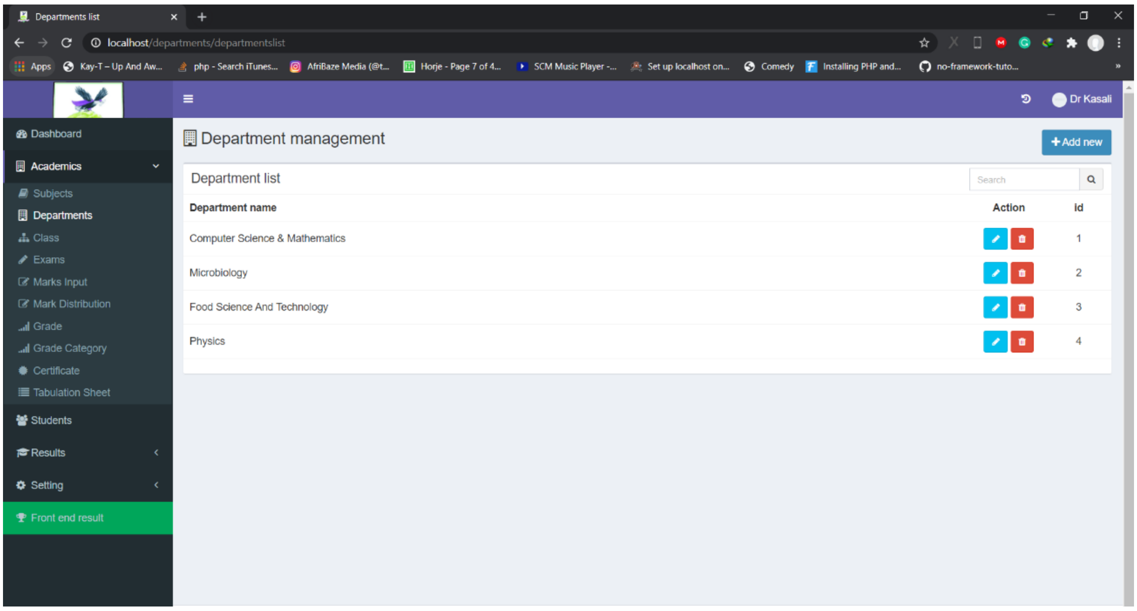
2020

| Student Roll | Student Name | Total Mark | GPA | Letter Grade |
|---|---|---|---|---|
| 16010301040 | Nwaobi Chukwuemeka Oluwatunmise | 60 | 0 | |
| 16010301011 | Olusegun Olayinka | 80 | 0 | |
| 16010301021 | Fadumo Ademola | 0 | 0 | |

**Figure 4.7 Broadsheet Page (Admin)**

### v) Departments

In this interface, the Exam Officer is tasked with managing the student's departments in the university. The Exam officer can add departments and delete students from the system.



Department management

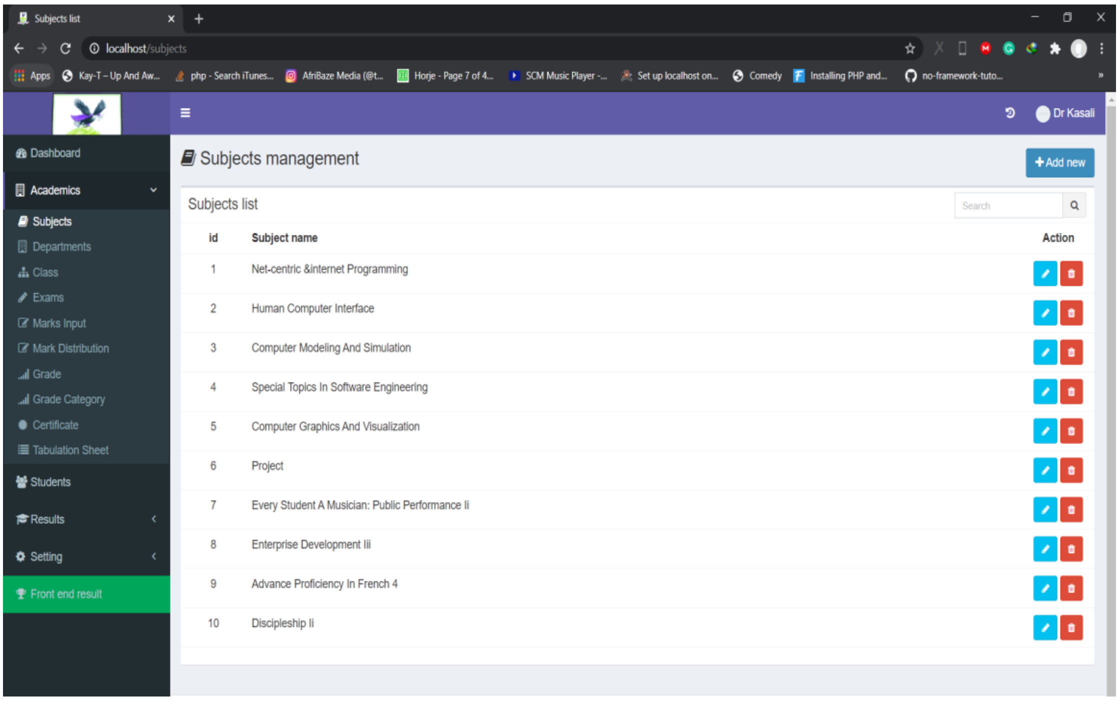
Department list

| Department name | Action | id |
|---|---|---|
| Computer Science & Mathematics | | 1 |
| Microbiology | | 2 |
| Food Science And Technology | | 3 |
| Physics | | 4 |

**Figure 4.8 Department Page (Admin)**

### vi) Courses

In this interface, the Exam Officer is tasked with managing the courses offered by each department in the university. The Exam officer can add courses and delete from the system.



Subjects management

Subjects list

| id | Subject name | Action |
|---|---|---|
| 1 | Net-centric &internet Programming | |
| 2 | Human Computer Interface | |
| 3 | Computer Modeling And Simulation | |
| 4 | Special Topics In Software Engineering | |
| 5 | Computer Graphics And Visualization | |
| 6 | Project | |
| 7 | Every Student A Musician: Public Performance Ii | |
| 8 | Enterprise Development Iii | |
| 9 | Advance Proficiency In French 4 | |
| 10 | Discipleship Ii | |

**Figure 4.9 Course Page (Admin)**

### 4.4.3 Lecturers Dashboard

The Lecturers dashboard shows the user interface which the Exam officer sees. The lecturer can upload students score sheet, edit profile and view student profiles.

### i) Homepage

This interface shows the lecturers homepage, where the lecturer can edit profile, view students' profile and submit student score sheets to the system.

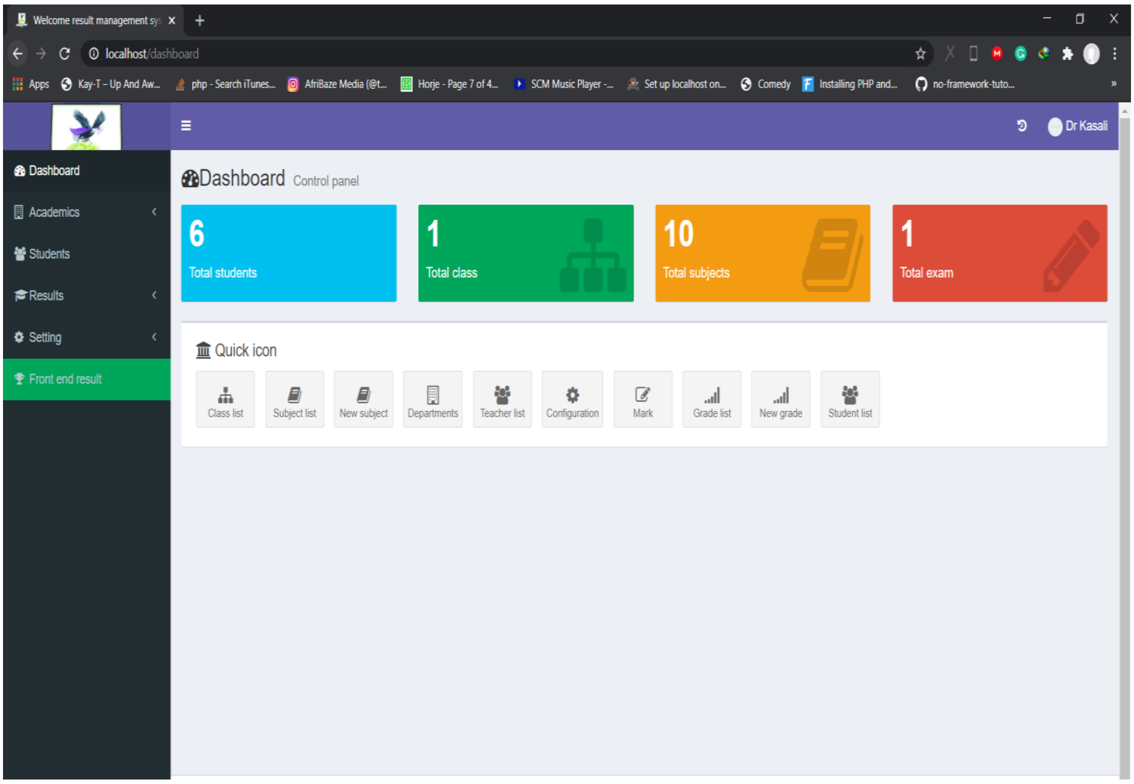
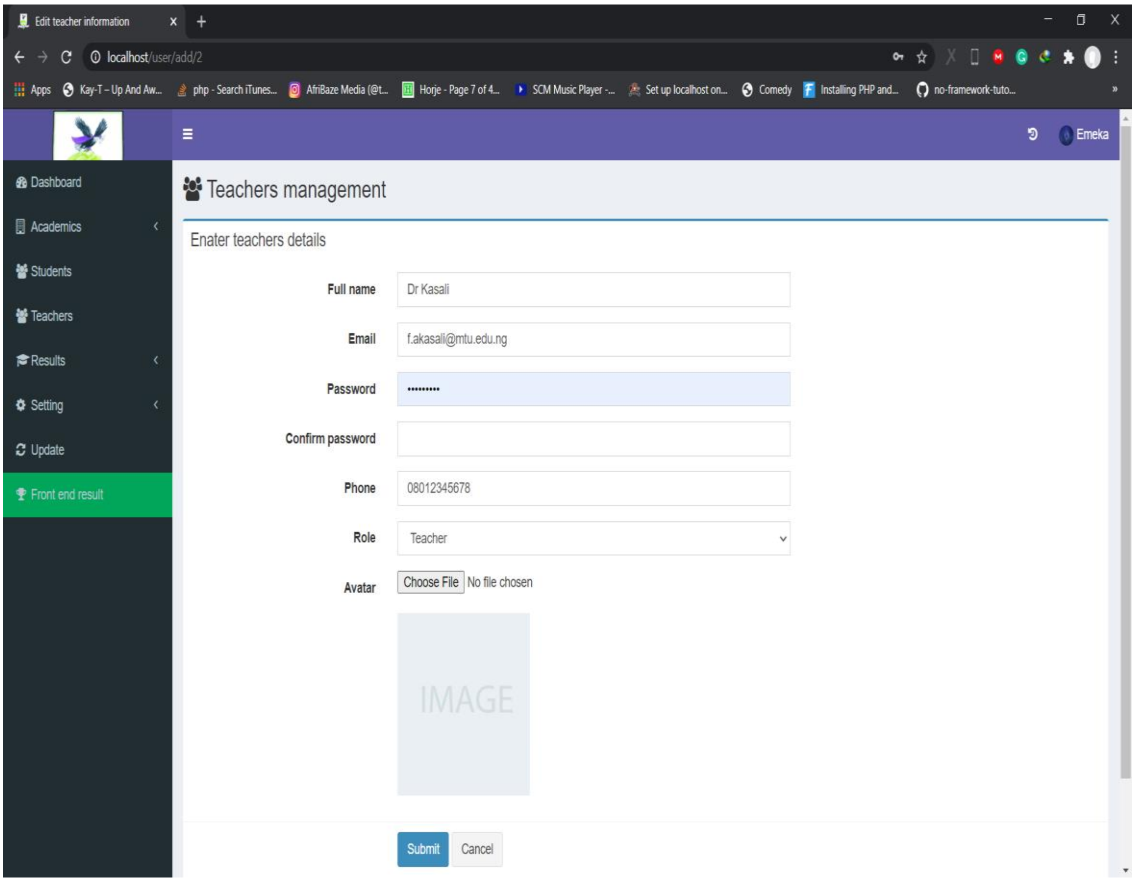


**Figure 4.10 Lecturer Dashboard (Lecturer)**

### ii) Edit Profile

This interface shows the lecturers page, where the lecturer can edit basic information about themselves.



Teachers management

Enater teachers details

Full name: Dr Kasali

Email: f.akasali@mtu.edu.ng

Password: •••••••••

Confirm password:

Phone: 08012345678

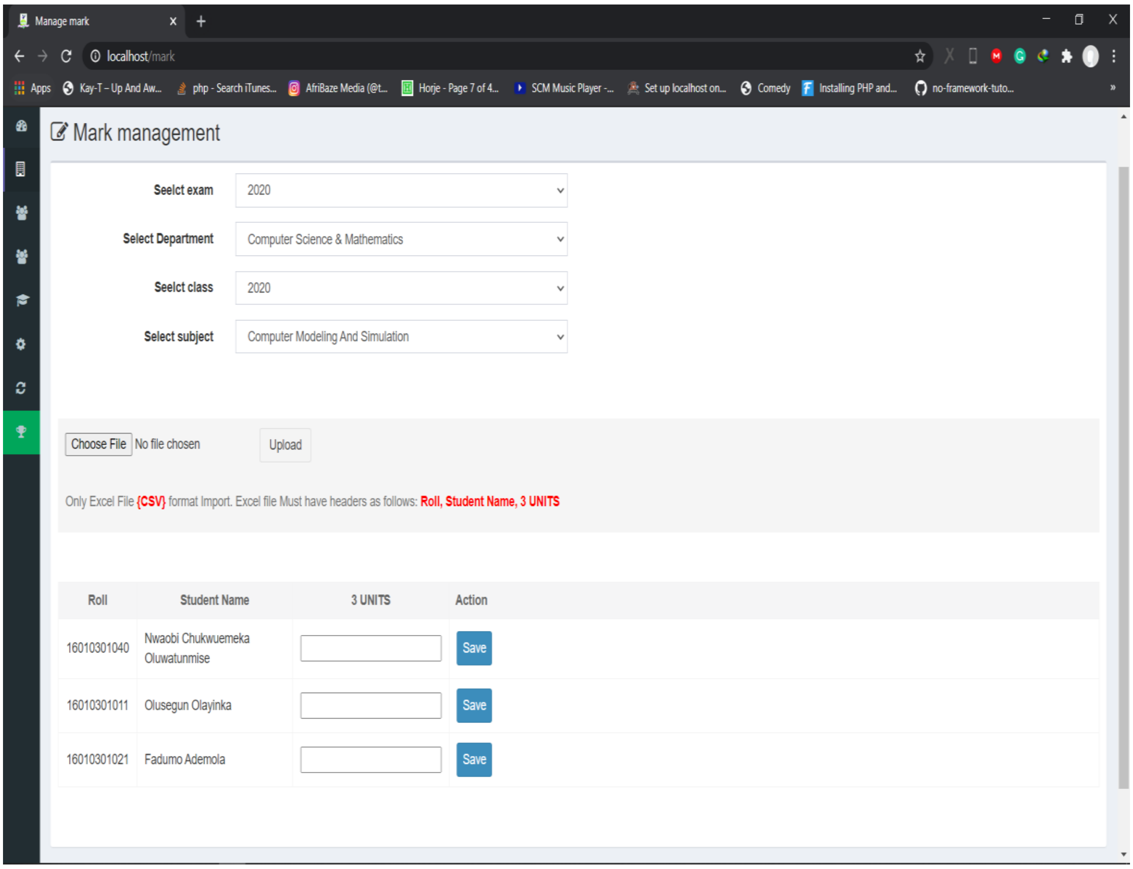
Role: Teacher

Avatar: No file chosen

Figure 4.11 Edit Profile Page (Lecturer)

### iii) Upload Result Score Sheet

This interface shows the page, where the lecturer selects the year, course, department and level he/she wants to upload the result score sheet to the system.



Mark management

| | |
|---|---|
| Seelct exam | 2020 |
| Select Department | Computer Science & Mathematics |
| Seelct class | 2020 |
| Select subject | Computer Modeling And Simulation |

Only Excel File **{CSV}** format Import. Excel file Must have headers as follows: **Roll, Student Name, 3 UNITS**

| Roll | Student Name | 3 UNITS | Action |
|---|---|---|---|
| 16010301040 | Nwaobi Chukwuemeka Oluwatunmise | | Save |
| 16010301011 | Olusegun Olayinka | | Save |
| 16010301021 | Fadumo Ademola | | Save |

**Figure 4.12 Upload Result Scoresheet Page (Lecturer)**

### 4.4.4 Students Dashboard

In this interface, the student can edit their profile, register for courses offered by the department and view results.

#### i) Homepage

This interface shows the students homepage, where the student can edit profile, have their course registration and view results.

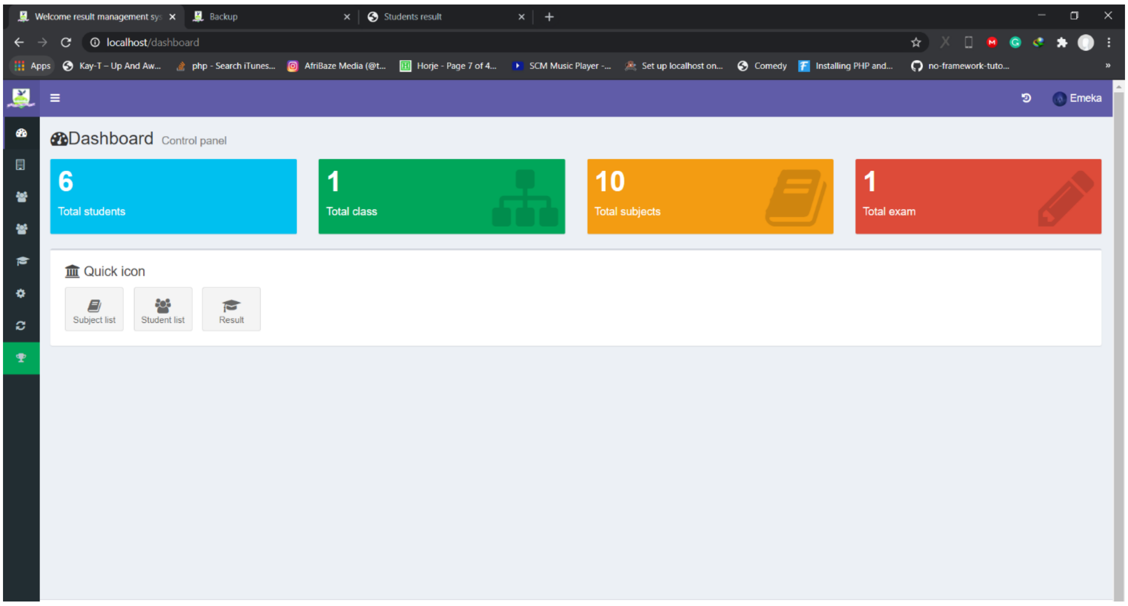
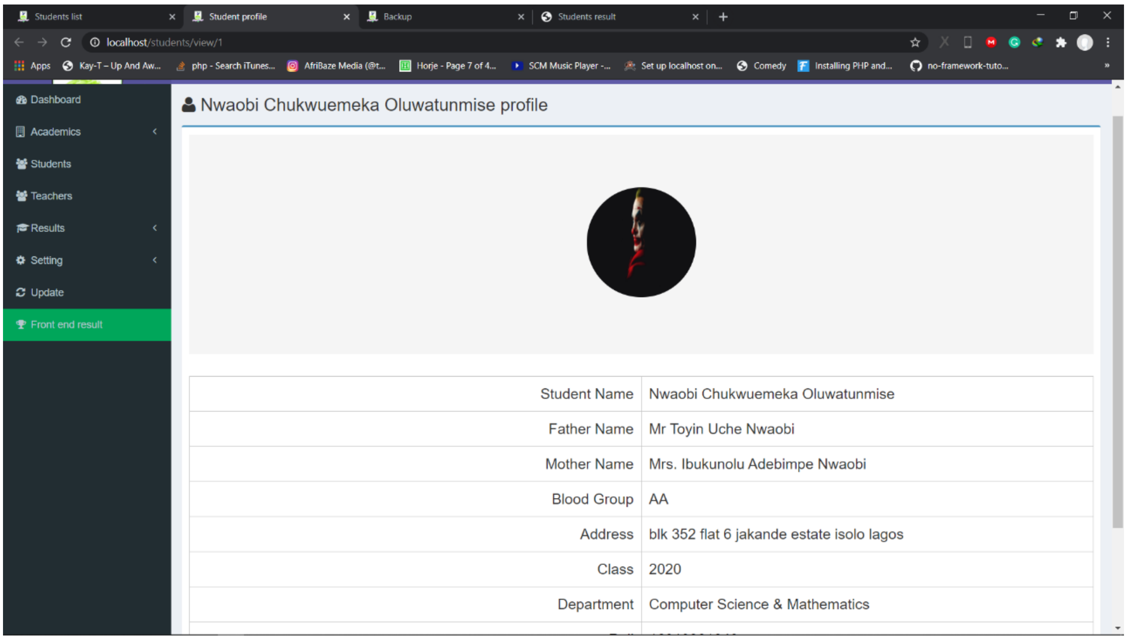


**Figure 4.13 Homepage (Student)**

### ii) View and Edit Profile

This interface shows the students profile, where the student can edit profile and update information about them.



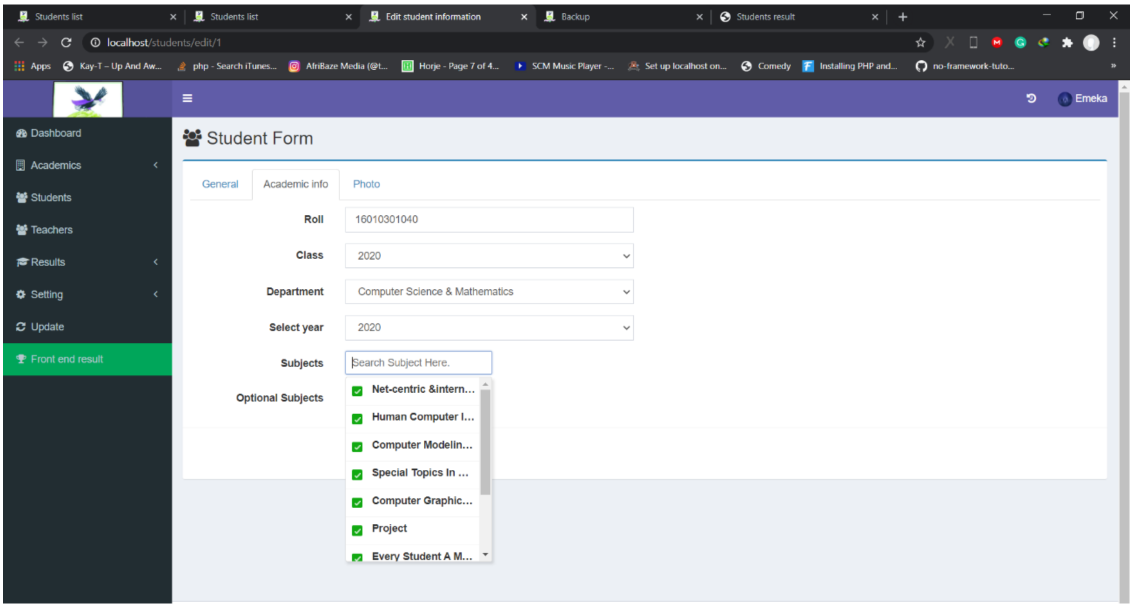
Nwaobi Chukwuemeka Oluwatunmise profile

| | |
|---|---|
| Student Name | Nwaobi Chukwuemeka Oluwatunmise |
| Father Name | Mr Toyin Uche Nwaobi |
| Mother Name | Mrs. Ibukunolu Adebimpe Nwaobi |
| Blood Group | AA |
| Address | blk 352 flat 6 jakande estate isolo lagos |
| Class | 2020 |
| Department | Computer Science & Mathematics |

**Figure 4.14 View and Edit Profile Page (Student)**

### iii) Courses Registration

This interface shows the how the students would register for the compulsory and elective courses to be taken for the semester. It also alerts the student on the particular outstanding courses he/she must take before taking the next course.



Student Form

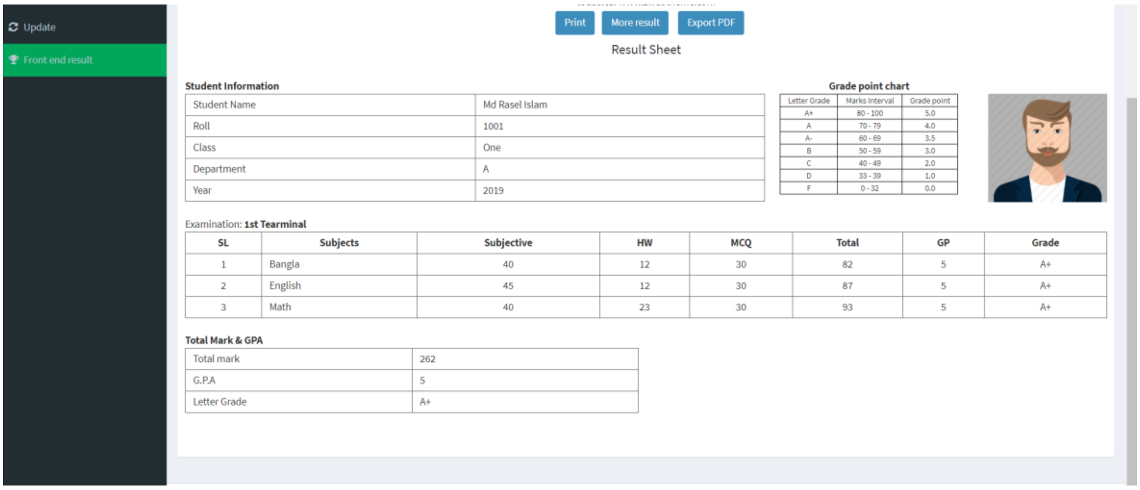
| | |
|---|---|
| Roll | 16010301040 |
| Class | 2020 |
| Department | Computer Science & Mathematics |
| Select year | 2020 |
| Subjects | Search Subject Here. |
| Optional Subjects | |

- ☑ Net-centric &intern...
- ☑ Human Computer I...
- ☑ Computer Modelin...
- ☑ Special Topics In ...
- ☑ Computer Graphic...
- ☑ Project
- ☑ Every Student A M...

**Figure 4.15 Course Registration Page (Student)**

### iv) View Results

This interface allows the student to view all the results obtained in the university, the result shows the Grade Point Average (GPA) obtained for the semester and the students current Cumulative Grade Point Average (CGPA) and it also notifies the student if the student is on a Wrong Standing or Good standing.



Result Sheet

**Student Information**

| | |
|---|---|
| Student Name | Md Rasel Islam |
| Roll | 1001 |
| Class | One |
| Department | A |
| Year | 2019 |

**Grade point chart**

| Letter Grade | Marks Interval | Grade point |
|---|---|---|
| A+ | 80 - 100 | 5.0 |
| A | 70 - 79 | 4.0 |
| A- | 60 - 69 | 3.5 |
| B | 50 - 59 | 3.0 |
| C | 40 - 49 | 2.0 |
| D | 33 - 39 | 1.0 |
| F | 0 - 32 | 0.0 |

Examination: **1st Tearminal**

| SL | Subjects | Subjective | HW | MCQ | Total | GP | Grade |
|---|---|---|---|---|---|---|---|
| 1 | Bangla | 40 | 12 | 30 | 82 | 5 | A+ |
| 2 | English | 45 | 12 | 30 | 87 | 5 | A+ |
| 3 | Math | 40 | 23 | 30 | 93 | 5 | A+ |

**Total Mark & GPA**

| | |
|---|---|
| Total mark | 262 |
| G.P.A | 5 |
| Letter Grade | A+ |

**Figure 4.16 View Result Page (Student)**

# CHAPTER FIVE

# SUMMARY, CONCLUSION AND RECOMMENDATIONS

## 5.0 Summary

Having reviewed the challenges with securing the result management system in the university, this proposed system is believed to help eradicate the problems associated with the unauthorized method by the introduction of a Biometric system which will make security more efficient and effective for the university's result management. The Result management system adds amazing values to the lives of both staff and lecturers as it aids in reduction of workload and reduction of data error. It can capture data, store, and view, add and delete records into the database when required. On the long run during the development of this system, some challenges where encountered, some of them includes it not being responsive on mobile and tablet devices

## 5.1 Contribution to Knowledge

The main contribution of knowledge was the ability to implement an Automated Broadsheet and Result management system using the Laravel Framework. With the help of Laravel, the system is protected from serious security risks like cross site request forgery, SQL injection, and cross site scripting.

## 5.2 Limitations

i. The system is not user friendly on tablet and mobile devices

ii. Due to limited data, the sending of broadsheets automatically via emails could not be implemented.

## 5.3 Recommendation for Further Study

It is recommended that the university's current result system should include Two (2) Factor Authentication system where the system sends a code to only registered emails to grant permission into the system. The following are also recommended.

i. Other types of authentication methods (Single-factor, Two-factor, Multi-factor authentication) can be used for securing result management systems.

ii. Other types of frameworks (AngularJS, Symfony, ASP.NET, Vue Js) can be used in building an Automated broadsheet and A Secured Result Management system.

iii. Other types of database management systems can be used by Result management systems.

## 5.4 Conclusion

This project takes a look at the various problems associated with the existing system which is the system not having enough security, time wastage amongst others. With all these problems being critically analyzed, a solution was embarked on, to eliminate these problems. With the design of an automated broadsheet and result management system such problems are considered to be eradicated.

In conclusion an automated broadsheet and result management system not only provides an opportunity to the university to enhance their result management system, but also can increase the profitability of the organization. This would improve the response because it automates the process of collecting, collating and retrieving student's information.

# REFERENCES

NUC (2010). National Grading Systems in Nigerian Universities retrieved from

https://www.nuc.edu.ng/stakeholders-review-grading-system-in-nigerian-universities/

Tutorialspoint (2019). Laravel Tutorial. Retrieved from https://www.tutorialspoint.com/laravel/

Cumulative Grade Point Average (2018). NUC wants universities to use five-point grading system Retrieved From

https://campusreporter.ng/nuc-wants-universities-to-use-five-point-grading-system/

Codecanyon. (2018.). Result Management System. Retrieved from https://codecanyon.net/item/result-management-system/22462424

Tutorialspoint. (2019). Laravel Tutorial. Retrieved from https://www.tutorialspoint.com/laravel/

Ajay and Abhishek (2012) Automated Students' Results Management Information System (SRMIS) retrived from https://www.researchgate.net/publication/333805147_Automated_Students'_Results_Management_Information_System_SRMIS

Tech Target (2020). What is a Use Case Retrieved from https://searchsoftwarequality.techtarget.com/definition/use-case

Okonigene (2008) Automated Students Result Management System retrieved from https://www.iiste.org/Journals/index.php/JIEA/article/download/3741/3790

Vecchioli (1999) The Design and Implementation of Student Academic Record Management System Retrieved From

http://maxwellsci.com/print/rjaset/v3-707-712.pdf

Barrett (1999) Automated Students Result Management System Retrieved From https://projects.ng/project/development-of-a-student-online-result-management-system/

Chidinma (2015) Automated Students Result Management System retrieved from https://www.iiste.org/Journals/index.php/JIEA/article/download/3741/3790

Ekanem and Ozuomba (2017) Development of Students Result Management System: A case study of University of Uyo retrieved from https://www.researchgate.net/publication/336242893_Development_of_Students_Result_Management_System_A_case_study_of_University_of_Uyo

Ghatge, M. S. T., & Bhoite, S. D. (2016). Improving College Working Efficiency Through Design and Development of A Computerized Information System (CIS). International Journal of Scientific Research, 4(7), 521-524.

Morningsunit University Result Management System retrieved from https://codecanyon.net/item/urms-university-result-management-system/19573557

# APPENDIX

## SOURCE CODE

```
<?php if(!defined('BASEPATH')) exit('No direct script access allowed');


class Login extends CI_Controller
{


    public function __construct()
    {
        parent::__construct();
        $this->load->model('login_model');
    }


    /**
    ** Index
    **/
    public function index()
    {
        $this->isLoggedIn();
    }


    /**
    ** This function used to check the user is logged in or not
    **/
    function isLoggedIn()
    {
        $isLoggedIn = $this->session->userdata('isLoggedIn');
        if(!isset($isLoggedIn) || $isLoggedIn != TRUE){
            $this->load->view('user/login');
        }else{
            redirect('/dashboard');
        }
    }



    /**
    ** This function used to logged in user
    **/
    public function loginMe()
    {
        // Get form validation
        $this->load->library('form_validation');
        $this->form_validation->set_rules('email',                                'Email',
'required|valid_email|max_length[128]|trim');
        $this->form_validation->set_rules('password',                          'Password',
'required|max_length[32]');

        if($this->form_validation->run() == FALSE){
            $this->index();
        }else{
```

```
// Get collection form data
$email    = $this->security->xss_clean($this->input->post('email'));
$password = $this->input->post('password');
$result   = $this->login_model->loginMe($email, $password);

if(!empty($result))
{
    $lastLogin = $this->login_model->lastLoginInfo($result->userId);

    // Set session data
    $sessionArray = array(
        'userId'     => $result->userId,
        'role'       => $result->roleId,
        'roleText'   => $result->role,
        'name'       => $result->name,
        'lastLogin'  => $lastLogin->createdDtm,
        'isLoggedIn' => TRUE
    );

    $this->session->set_userdata($sessionArray);
    unset($sessionArray['userId'], $sessionArray['isLoggedIn'], $sessionArray['lastLogin']);

    // Set login info
    $loginInfo = array(
            "userId"       => $result->userId,
            "sessionData"  => json_encode($sessionArray),
            "machineIp"    => $_SERVER['REMOTE_ADDR'],
            "userAgent"    => getBrowserAgent(),
            "agentString"  => $this->agent->agent_string(),
            "platform"     => $this->agent->platform()
        );

        $this->login_model->lastLogin($loginInfo);
        redirect('/dashboard');
      }else{
        $email_pass_mismatch = getlang('email_pass_mismatch');
        $this->session->set_flashdata('error', $email_pass_mismatch);
        redirect('login');
      }
   }
}

/**
** This function used to load forgot password view
**/
public function forgotpassword()
{
   $isLoggedIn = $this->session->userdata('isLoggedIn');
   if(!isset($isLoggedIn) || $isLoggedIn != TRUE){
```

```
            $this->load->view('user/forgotpassword');
        }else{
            redirect('/dashboard');
        }
    }

    /**
    ** This function used to generate reset password request link
    **/
    function resetPasswordUser()
    {
        $status = '';
        $this->load->library('form_validation');
        $this->form_validation-
>set_rules('login_email','Email','trim|required|valid_email');
        if($this->form_validation->run() == FALSE){
            $this->forgotPassword();
        }else{
            $email = $this->security->xss_clean($this->input->post('login_email'));
            if($this->login_model->checkEmailExist($email))
            {
                $encoded_email = urlencode($email);
                $this->load->helper('string');
                $data['email']        = $email;
                $data['activation_id'] = random_string('alnum',15);
                $data['createdDtm']    = date('Y-m-d H:i:s');
                $data['agent']         = getBrowserAgent();
                $data['client_ip']     = $this->input->ip_address();
                $save = $this->login_model->resetPasswordUser($data);

                if($save){
                  $data1['reset_link'] = base_url() . "resetPasswordConfirmUser/" .
$data['activation_id'] . "/" . $encoded_email;
                  $userInfo = $this->login_model->getCustomerInfoByEmail($email);

                  if(!empty($userInfo)){
                    $reset_pass = getlang('reset_pass');
                    $data1["name"] = $userInfo[0]->name;
                    $data1["email"] = $userInfo[0]->email;
                    $data1["message"] = $reset_pass;
                  }

                  $sendStatus = resetPasswordEmail($data1);

                  if($sendStatus){
                    $pass_link_sent = getlang('pass_link_sent');
                    $status = "send";
                    setFlashData($status, $pass_link_sent);
                  } else {
                    $email_failed = getlang('email_failed');
```

```
                    $status = "notsend";

                    setFlashData($status,  $email_failed);

                }

            }else{

                $sent_details_error = getlang('sent_details_error');

                $status = 'unable';

                setFlashData($status, $sent_details_error);

            }

        }else{

            $email_registered = getlang('email_registered');

            $status = 'invalid';

            setFlashData($status, $email_registered);

        }

        redirect('forgotpassword');

    }

}


/**

** This function used to reset the password

** @param string $activation_id : This is unique id

** @param string $email : This is user email

**/

function resetPasswordConfirmUser($activation_id, $email)

{

    // Get email and activation code from URL values at index 3-4
```

```
        $email = urldecode($email);

        // Check activation id in database
        $is_correct    =    $this->login_model->checkActivationDetails($email,
$activation_id);

        $data['email']       = $email;
        $data['activation_code'] = $activation_id;

        if ($is_correct == 1){
            $this->load->view('user/newPassword', $data);
        }else{
            redirect('login');
        }
    }

    /**
    ** This function used to create new password for user
    **/
    function createPasswordUser()
    {
        $status     = '';
        $message    = '';
        $email      = $this->input->post("email");
        $activation_id = $this->input->post("activation_code");
        $this->load->library('form_validation');
        $this->form_validation-
>set_rules('password','Password','required|max_length[20]');
        $this->form_validation->set_rules('cpassword','Confirm
Password','trim|required|matches[password]|max_length[20]');

        if($this->form_validation->run() == FALSE){
            $this->resetPasswordConfirmUser($activation_id, urlencode($email));
        }else{
            $password  = $this->input->post('password');
            $cpassword = $this->input->post('cpassword');

            // Check activation id in database
            $is_correct = $this->login_model->checkActivationDetails($email,
$activation_id);

            if($is_correct == 1)
            {
                $this->login_model->createPasswordUser($email, $password);
                $status  = 'success';
                $message = getlang('pass_success');
            }else{
                $status  = 'error';
                $message = getlang('pass_error');
            }


            setFlashData($status, $message);

            redirect("login");

        }

    }

}


?>
```