

**DESIGN AND IMPLEMENTATION OF A WEB BASED
SESSIONAL RESULT COMPUTATION SYSTEM
CASE STUDY OF UNIVERSITY OF BENIN**

BY

**OKOLI – UME CHIOMA
(PSC1611591)**

**DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF PHYSICAL SCIENCES
UNIVERSITY OF BENIN,
BENIN CITY,
EDO STATE, NIGERIA.**

NOVEMBER, 2023.

**DESIGN AND IMPLEMENTATION OF A WEB BASED
SESSIONAL RESULT COMPUTATION SYSTEM
CASE STUDY OF UNIVERSITY OF BENIN**

BY

OKOLI-UME CHIOMA

(PSC1611591)

**THE PROJECT SUBMITTED TO THE DEPARTMENT OF
COMPUTER SCIENCE, FACULTY OF PHYSICAL
SCIENCES, UNIVERSITY OF BENIN, BENIN CITY.
IN PARTIAL FULFILMENT OF THE REQUIREMENT
FOR THE AWARD OF BACHELOR OF SCIENCE (B.Sc)
DEGREE IN COMPUTER SCIENCE.**

NOVEMBER, 2023.

CERTIFICATION

This is to certify that this project was carried out by **OKOLI-UME CHIOMA** with matriculation number **PSC1611591** of the Department of Computer Sciences, faculty of physical sciences, University of Benin, under the supervision of **Prof. V.I. OSUNBOR**

.....
Prof. V. I. Osubor
(Project Supervisor)

.....
Date

APPROVAL

This project is hereby approved by the Department of Computer Science, Faculty of Physical Science, university of Benin, Benin City in partial fulfillment of the requirement for the award of Bachelor of Science (B.Sc.) Degree in Computer science.

Prof. (Mrs) V.I Osubor
(Project Supervisor)

Date

Prof. G.O Ekuobase
(Head of Department)

Date

DEDICATION

This project is dedicated to God Almighty.

ACKNOWLEDGMENT

My sincere gratitude goes to God Almighty for his Grace and infinite Mercy towards me during the period of my studies in the University of Benin.

I wish to appreciate the effort and encouragement of my project supervisor, Prof. V.I. Osubor who went through my work to offer relevant corrections to bring it to baseline. Your corrections and truthful appetite for excellence motivated me to be more precise. May God Almighty richly bless and reward you and also meet you at the point of your needs, thank you very much ma, I am very grateful.

My sincere gratitude goes to all the lecturers of Computer Science whom I sat under their teaching during my stay in this department. Worthy note include HOD, Prof. G.O. Ekuobase, Prof. V.I. Osubor, Prof. F.I. Amadin, Prof. (Mrs.) R.O. Egwali, Prof. A.A. Imiavian, Prof. Chete, Prof. Koyeha, Mr. S.O.P. Oliomogbe, Dr. (Mrs.) R.O Osasere, Mr. Otikiti, Mrs. Usiobaifo, Miss Usiosefe, Dr. Nweli, Dr. Idehen, Dr. G.Aziken, Mr. F.O. Oliha, Dr. E.C. Igodan, Dr. Princewill, others too numerous to mention, and not to forget all members of the non academic staff of the department for their support and encouragement, May God grant all your heart desires and reward you for your prodigious contributions to the success of this programme.

I wish thank my parents MR. and Mrs. Samuel Umeobi and also my wonderful siblings for their encouragement, prayers and financial support throughout this program, may Almighty God bless and reward you.

And also my friends who has been there for me and always encouraging me to be the best version of myself and also push me to take my studies serious, may God bless you and grant your heart desires.

Lastly my appreciation to everyone who has helped me in one way or the other towards the success of this program may Almighty God blesses you all.

TABLE OF CONTENTS

Title Page	-	-	-	-	-	-	-	-	-	ii
Certification-	-	-	-	-	-	-	-	-	-	iii
Approval -	-	-	-	-	-	-	-	-	-	iv
Dedication-	-	-	-	-	-	-	-	-	-	v
Acknowledgements-	-	-	-	-	-	-	-	-	-	vi
Table of Contents-	-	-	-	-	-	-	-	-	-	vii
List of Table	-	-	-	-	-	-	-	-	-	viii
List of Figures-	-	-	-	-	-	-	-	-	-	x
Abstract-	-	-	-	-	-	-	-	-	-	xi

CHAPTER ONE

INTRODUCTION

1.1 Background of The Study.	-	-	-	-	-	-	-	-	1
1.1 Statements of The Problem	-	-	-	-	-	-	-	--	2
1.2 Aim and Objectives-	-	-	-	-	-	-	-	-	2
1.3 Significance of The Study -	-	-	-	-	-	-	-	-	3
1.4 Scope of The Study -	-	-	-	-	-	-	-	-	3
1.5 Limitations of The Study--	-	-	-	-	-	-	-	-	3
1.6 The Research Layout -	-	-	-	-	-	-	-	-	4
1.7 Definition of Terms -	-	-	-	-	-	-	-	-	4

CHAPTER TWO

LITERATURE REVIEW

2.1 Result Computation	-	-	-	-	-	-	-	-	6
2.2 Result Computation Techniques --	-	-	-	-	-	-	-	-	6
2.3 Review of Related Works -	-	-	-	-	-	-	-	-	7

2.4 Benefits of The Result Computation System	-	-	-	-	-	-	-	11
---	---	---	---	---	---	---	---	----

CHAPTER THREE

METHODOLOGY

3.1 System Architecture	-	-	-	-	-	-	-	13
3.2 System Components	-	-	-	-	-	-	-	13
3.3 Methodology Used	-	-	-	-	-	-	-	14
3.4 System Functionality	-	-	-	-	-	-	-	16
3.4.1 Flowchart of the Developed System	-	-	-	-	-	-	-	16
3.4.2 Methods of Data Collection	-	-	-	-	-	-	-	17
3.4.3 The Action Diagram	-	-	-	-	-	-	-	19
3.4.4 The Entity Relationship Diagram of The System	-	-	-	-	-	-	-	20
3.5 The Database Design	-	-	-	-	-	-	-	20

CHAPTER FOUR

IMPLEMENTATION

4.1 Features, Languages and Technologies Used	-	-	-	-	-	-	-	25
4.2 Design Specification	-	-	-	-	-	-	-	27
4.2.1 Home Page Design	-	-	-	-	-	-	-	27
4.2.2 The Login Design	-	-	-	-	-	-	-	27
4.2.3 Sign Up Page Design	-	-	-	-	-	-	-	29
4.2.4 The Admin Dashboard Page	-	-	-	-	-	-	-	29
4.3 System Requirements	-	-	-	-	-	-	-	34
4.4 Summary	-	-	-	-	-	-	-	34

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary - - - - - 35

5.2 Conclusion - - - - - 35

5.3 Recommendation - - - - - 36

References

Appendix

LIST OF TABLE

Table 1: Course Advisor Table	-	-	-	-	-	-	21
Table 2: Courses Table	-	-	-	-	-	-	21
Table 3 Password Reset Table	-	-	-	-	-	-	22
Table 4 Result Table- -	-	-	-	-	-	-	22
Table 5: Students Table -	-	-	-	-	-	-	23
Table 6: Students Courses Table-	-	-	-	-	-	-	23

LIST OF FIGURES

Figure 1: Agile Methodology Circle --	-	-	-	-	-	-	-	15
Figure 2: Flowchart diagram of the Developed System -	-	-	-	-	-	-	-	16
Figure 3: Use Case Diagram for the System -	-	-	-	-	-	-	-	18
Figure 4: The Action Diagram of the System -	-	-	-	-	-	-	-	19
Figure 5: The Entity Relationship Diagram of the System	-	-	-	-	-	-	-	20
Figure 6: The Home Page -	-	-	-	-	-	-	-	27
Figure 7: The Login Page	-	-	-	-	-	-	-	28
Figure 8: Password Reset -	-	-	-	-	-	-	-	28
Figure 9: The sign-Up Page --	-	-	-	-	-	-	-	29
Figure 10: The Admin Dashboard	-	-	-	-	-	-	-	30
Figure 11: The Admin Add Course Interface -	-	-	-	-	-	-	-	30
Figure 12: The Result Interfaces-	-	-	-	-	-	-	-	31
Figure 13: The Computational Result Interfaces -	-	-	-	-	-	-	-	31
Figure 14: The Computational Result Interface For A Student -	-	-	-	-	-	-	-	32
Figure 15: The Input Field Interface	-	-	-	-	-	-	-	32
Figure 16: The Result Page for a Student-	-	-	-	-	-	-	-	33

ABSTRACT

The web-based sectional result computation system abstracts the complexities of calculating and generating academic results in a University or educational institution. It provides a user-friendly interface that simplifies data management, computation algorithms, result generation, security, access control, and integration with other systems. By abstracting these components, the system streamlines the result computation process and ensures accuracy, reliability, and timeliness. It simplifies data storage, retrieval, and organization, automates grade calculations and performance indicators, and generates result sheets and reports in customizable formats. The system also abstracts security measures, ensuring authorized access to student results, and integrates with other institutional systems for seamless data synchronization. This abstraction allows users, such as faculty, staff, and administrators, to focus on result analysis, interpretation, and decision-making, rather than getting overwhelmed by technical details. The web-based nature of the system enables easy access from anywhere with an internet connection, enhancing accessibility and convenience. Overall, the result computation system's abstraction simplifies and automates result processing, improves efficiency, reduces errors, and provides standardized and reliable frameworks for handling academic outcomes. It empowers users with accurate and timely student results, facilitating informed decision-making and enhancing the overall academic experience within the institution.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

The evaluation of academic performance is a crucial aspect of educational assessment, providing students with valuable feedback on their progress and serving as a basis for determining their academic standing (Ukem and Onoyom-Ita, 2022). The conventional method of manually computing results is time-consuming and error-prone, particularly in universities with a high number of students and numerous courses (Añulika et al., 2023). There is a growing interest in the development of automated result computation systems for universities, aiming to streamline the process, reduce the workload on academic staff, and enhance accuracy and efficiency. These systems typically employ software programs and algorithms to analyze student performance data and compute results (Matemilayo et al., 2021).

Femi and Ijesunor (2022) noted that automated systems can handle substantial amounts of data and perform complex computations, such as grading schemes and curve-fitting, more accurately and consistently than traditional manual methods (Ekanem et al., 2021). An automated system's main advantage is providing real-time feedback to students, enabling them to monitor their academic progress and make informed decisions about their coursework. Additionally, it aids academic staff in identifying areas where students struggle, facilitating targeted interventions to improve student performance (Smith and Johnson, 2020).

In summary, the development of an automated result computation system for a university has the potential to enhance the accuracy, efficiency, and effectiveness of academic

assessment. By leveraging advanced software programs and algorithms, universities can streamline result computation, reduce the workload on academic staff, and offer real-time feedback to students on their academic progress (Mohamed and Mohamed, 2019).

1.1 STATEMENTS OF THE PROBLEM

The traditional manual method of computing academic results in universities is time-consuming and error-prone, particularly in large institutions with a high volume of students and courses (Femi and Ijesunor, 2022). This leads to result release delays and grading inconsistencies, causing dissatisfaction among students and academic staff. Moreover, the manual computation is resource-intensive, placing a significant workload on academic staff (Ukem and Onoyom-Ita, 2022), potentially leading to burnout and computation errors due to the sheer volume of data (Martinez et al., 2015).

Hence, there is a necessity to develop an automated result computation system overcoming the limitations of the traditional manual method. The automated system should handle large data volumes, compute results accurately and efficiently, and provide real-time feedback and analytics to both students and academic staff. Additionally, the system should be user-friendly for seamless integration with existing university systems and ensure a high level of security to protect student data.

1.2 AIM AND OBJECTIVES

The main aim of this project is to develop a web-based result computation portal for the University of Benin. The specific objectives are:

1. Develop a web-based result computation system portal for accurately and efficiently computing results for all university courses.

2. Design a user-friendly system utilizing web technologies like CSS and Bootstrap 5 to reduce the workload and enhance grading accuracy and consistency.
3. Implement security measures such as password integration.
4. Efficiently conduct the academic assessment process for a faster results release and streamlined processes for students and academic staff.
5. Provide a scalable system to accommodate the growing number of students and courses offered by the university.

1.3 SIGNIFICANCE OF THE STUDY

The development of a result computation portal holds significant implications for academic assessment, student experience, and the institution's overall reputation. By improving the accuracy and efficiency of the academic assessment process, an automated result computation system contributes to elevated academic standards, benefiting students and academic staff.

1.4 SCOPE OF THE STUDY

The scope includes the development of a web-based portal for computing and publishing results, its implementation for use by course advisors, and integration of features like exam results input, grading systems, and report generation.

1.5 LIMITATIONS OF THE STUDY

The limitations encompass limited access to necessary resources, a constrained time frame for development, testing, and debugging, as well as limited availability of data for testing and validation. User acceptance may also be limited due to factors like preferences, technical skills, and resistance to change.

1.6 THE RESEARCH LAYOUT

The arrangement of this research work and a concise overview of the chapters are outlined below:

- Chapter One: Introduction to the research work/study, laying the groundwork for the research.
- Chapter Two: Literature review, reviewing various methods and approaches used to address result computation challenges.
- Chapter Three: Methodology and dataset used in the study are detailed in this section.
- Chapter Four: Focuses on the setup, implementation, experimental results, and performance criteria of the methods applied in the research work.
- Chapter Five: The concluding stage of the research work, including recommendations for future endeavors.

1.7 DEFINITION OF TERMS

- Software: A set of instructions or programs designed for specific tasks on a computer system. It encompasses programs, associated documentation, data, and configuration files necessary for program operation. In contrast, a program is a specific set of instructions in a programming language to perform tasks on a computer system.
- Web Application: A software application accessed through a web browser over a network, such as the internet or an intranet. It follows a client-server model, where the client (typically a web browser) sends requests to the server, and the server processes and responds accordingly.

- Internet: A global network connecting computer systems and devices through standard communication protocols. It spans the globe, linking millions of devices, servers, and computers.
- Technology: The application of scientific knowledge, tools, and techniques to create practical solutions or enhance existing products, services, or processes. It involves the development, production, and utilization of tools, machines, systems, and techniques to solve problems or achieve specific goals.
- Portal: In software development, a web-based platform providing users a single point of access to various resources, services, and applications.
- HTTPS: Hypertext Transfer Protocol Secure, an extension of the HTTP protocol for secure communication over the internet. HTTPS combines the HTTP protocol with the SSL/TLS cryptographic protocol.

CHAPTER TWO

2.1 RESULT COMPUTATION

Result computation refers to the process of calculating and generating academic results for students based on their performance in a course or program. It involves considering various factors such as assignments, tests, projects, and final exams, and assigning grades or marks to students based on their performance in each area (Williams et al., 2016). Typically, result computation utilizes specialized software applications and databases designed for managing and processing academic results (Gupta and Sharma, 2018). The process may be either automated or semi-automated, depending on the grading system's complexity and the number of students involved (Chen and Li, 2017). The accuracy and timeliness of result computation are crucial for students' academic success and the overall efficiency of the academic institution. It must be conducted transparently and fairly, considering all relevant factors and adhering to established grading policies and procedures (Brown et al., 2019).

2.2 RESULT COMPUTATION TECHNIQUES

Result computation involves several key techniques, including:

1. **Weighted Average:** This technique, employing weighted averages to compute grades, is a common approach in many educational institutions.
2. **Absolute Grading:** Absolute grading systems, with varied grading scales and criteria, have been in use for many years. In the United States, the A to F grading scale was established in the early 20th century (Kim et al., 2017).
3. **Relative Grading:** Introduced by educational psychologist Edward Thorndike in the early 20th century, this technique proposes assigning grades based on the

distribution of scores among students rather than an absolute standard (Johnson et al., 2018).

4. Norm-Referenced Grading: Originating from the work of educational psychologist Lee Cronbach in the mid-20th century, this technique compares students' performance against a larger group of test-takers or a normative sample (Adams et al., 2019).
5. Criterion-Referenced Grading: First introduced by educational psychologist Robert Glaser in the 1960s, this technique suggests grading based on the achievement of specific learning outcomes rather than a comparison with other students (Williams et al., 2016).

2.3 REVIEW OF RELATED WORKS

Añulika et al. (2023) developed a computer software application to automate results processing. The software, created using PHP and MYSQL, proved effective in handling tasks such as product registration and report generation. The system eliminated the need for manual score recording, calculations, and result booklet printing, making the teacher's job more efficient. Ukem and Onoyom-Ita (2022) employed the Waterfall Software Life Cycle model to develop a network-capable database. Femi and Ijesunor (2022) reported successful results with the use of MYSQL, Dreamweaver, PHP, and JavaScript. The system provided downloadable reports, catering to the result processing needs of public secondary schools. Matemilayo et al. (2021) developed a software application using HTML5, CSS8, and Java Script for client-side, PHP for server-side, and MYSQL for the database. The system demonstrated efficiency in computing Grade Point Average (GPA)

and Cumulative Grade Point Average (CGPA), offering benefits such as cost reduction and increased accuracy and efficiency.

Ekanem et al. (2021) conducted a study on an automated platform designed for managing the results of all categories of students in a seamless and interactive manner. The system was developed using PHP, CSS, HTML, and MYSQL, and it was hosted locally through the Apache web server. The chosen software development methodology was a Participatory Incremental Process Model (PIP). Data for the study were obtained from the Department of Electrical/Electronic and Computer Engineering at the University of Uyo. The paper provides a functional decomposition of the system, outlining its key modules to explain the major functionalities it offers. Additionally, a use case diagram is presented, illustrating the different categories of system users and the various functionalities associated with each user. The study includes screenshots demonstrating various system functionalities during the test run of the functional system.

Brailas et al. (2020) facilitated collective reflective practice in a learning group within an academic setting. The technical infrastructure of the system was fully implemented using free web services, requiring minimal technical knowledge or expertise. A crucial technical component of the proposed system was a web mash-up hub node that filtered and aggregated relevant posts from participating members' blogs. The overall design process, including theoretical inspiration, context, and implementation, is presented in full detail.

Eludire (2017) made commendable progress in developing a software system for student registration and matriculation number generation, providing an organized and efficient approach to managing student information. The system's ability to generate both pre-

defined and customized reports enhanced administrative decision-making processes. However, challenges related to networked deployment and limitations on the number of records and file sizes called for further attention and improvement. This research serves as a valuable contribution to the field of student registration and matriculation systems, shedding light on both achievements and areas for refinement. As technology continues to advance, future studies may build upon Eludire's work to create even more robust and scalable systems, ultimately benefiting educational institutions in streamlining administrative tasks and improving overall efficiency.

Raphael (2022) conducted research to develop a web-based Result Management System aimed at computing and managing student results. The system allowed both students and administrators to register and use its functionalities. The term "admin" in this context refers to individuals such as the Head of Department (H.O.D), Dean of Faculty, or Course Advisor, providing flexibility in its usage. The Result Management System successfully addressed the challenge of effectively storing student results. However, it had certain limitations. One significant limitation was that the system did not offer an automated way of computing results. Instead, results still needed to be calculated manually. While the system included a GPA calculator, it was not fully automated, which may have affected its overall efficiency and effectiveness.

Ekpenyong (2018) conducted research to develop an Intelligent Knowledge-Based System (IKBS) using the Microsoft Excel spreadsheet program. The system leveraged the programming facilities offered by Excel, allowing for efficient monitoring of student performance through cell referencing. This tracking included total scores, grades, cumulative points, and other relevant information, enhancing the process of result

computation. The IKBS proved to be a valuable tool in improving result computation. However, it also exhibited some drawbacks. The major limitation of Ekpenyong's work was that the programming was hard-coded into the cells. This approach limited the system's flexibility, making it challenging to adapt for different scenarios or modify without substantial programming expertise. As a result, customization and modifications required a high level of technical skill, potentially making it less user-friendly for individuals without extensive programming knowledge.

Choji and Ezenma (2015) undertook research to develop a software system utilizing the PHP programming language and MySQL database management system. The software aimed to streamline administrative tasks by allowing staff members to register and log in using default usernames and passwords. Once logged in, staff could enter students' raw scores for the subjects they taught, view submitted results, and change their passwords as needed. The system's structure designated the principal as the system administrator, granting them access to view staff, students, course registrations, and any student's results irrespective of their class. To ensure user-friendliness, the software employed a graphical user interface (GUI), making it accessible and easy to use. The utilization of a relational database system further ensured data integrity and organization. Notably, this software stood out from other available systems as it combined both registration and result computation functionalities. While many systems focused on either one of these aspects, this software encompassed both, providing a comprehensive solution for administrative tasks in an educational setting. However, the provided dataset did not offer specific details on additional limitations of the system beyond the mentioned functionalities.

Further research may be required to explore any potential drawbacks or limitations not covered in the available information.

From the literature review, the system will focus on developing a result computation system that works in conjunction with a course registration system utilizing the same database. The result computation system is specifically designed for course advisors, enabling them to add courses that students can subsequently register for through the course registration system. Once students have registered, course advisors can proceed to compute the results for those courses using the result computation system. In response to the limitations associated with the manual management of student results, we propose a solution in the form of a web-based application that automates the result processing. The development of this web application utilized the PHP scripting language, known as Hypertext Preprocessor, while the underlying database management system employed is MySQL, a reliable relational database management system.

2.4 BENEFITS OF THE RESULT COMPUTATION SYSTEM

The advantages of implementing a web-based result computation system for a university are significant and should not be underestimated. Key benefits include:

1. **Streamlined Result Processing:** The system automates the result computation process, reducing manual errors and saving valuable time for faculty and staff.
2. **Increased Efficiency:** By automating tasks such as grade calculation and GPA computation, the system improves efficiency and reduces the workload on faculty members and administrative staff.

3. **Enhanced Accuracy and Data Integrity:** The digital system eliminates the risks of human errors and inconsistencies associated with manual result computation, ensuring accurate and reliable results.
4. **Timely Result Publication:** The automated system expedites the result computation process, allowing for the timely publication of results to students, reducing anxieties and providing a better overall experience.
5. **Customized Reporting:** The system offers customizable reports, enabling faculty members and administrators to generate specific result reports based on their requirements, making data analysis more effective and tailored to their needs.
6. **Integration with Existing Systems:** The system can integrate with existing student information systems, providing a seamless flow of data and reducing the need for duplicate data entry.
7. **Long-term Cost Savings:** While initial implementation costs are involved, the long-term benefits and savings from reduced manual efforts, improved efficiency, and minimized errors outweigh the investment made in the system.

CHAPTER THREE

METHODOLOGY

3.1 SYSTEM ARCHITECTURE This web-based result computation system adopts a client-server architecture. The client serves as the user interface for course advisors, while the server manages user requests and result generation. The server, hosted on a shared platform for constant availability and scalability, utilizes the Hostinger free hosting service, specifically 000webhost. PHP, a back-end programming language, is employed to implement the server, which interacts with a MySQL database for data storage and retrieval.

On the client side, a web application is developed using HTML, CSS, JavaScript, and Bootstrap 5 – a popular CSS framework for constructing user interfaces. The web application communicates with the server through RESTful API methods, including HTTP methods for sending user requests and receiving responses.

3.2 SYSTEM COMPONENTS The web-based result computation system comprises the following components:

1. **User Interface:** The front-end component facilitates user interaction, providing a user-friendly interface for course advisors to input data and view results.
2. **REST API:** Responsible for communication between the client and server, handling requests from the client and sending back responses.

3. **Server:** Processes user requests and generates results, interacting with the database for data storage and retrieval.
4. **Database:** Stores all necessary system data, including student and teacher information, course data, and result data.

3.3 METHODOLOGY USED

The project employs the Agile Methodology, an iterative and incremental approach to software development emphasizing flexibility, collaboration, and continuous improvement. The Agile methodology aims to deliver working software in short iterations or sprints, typically lasting from one to four weeks, guided by a set of values and principles outlined in the Agile Manifesto.

Key aspects of Agile methodology include:

1. **Iterative and Incremental Development:** Projects are divided into small iterations or sprints, producing a working increment of the software to allow for regular feedback and adaptation based on changing requirements.
2. **Adaptive Planning:** Agile projects embrace changing requirements, focusing on high-level planning initially and adapting plans as insights and feedback accumulate during the project.
3. **Cross-functional Teams:** Collaboration among team members from different disciplines (developers, testers, designers, etc.) is promoted for effective communication and collective ownership of the software's quality and success.
4. **Customer Collaboration:** Frequent interaction and collaboration with the customer or product owner are encouraged to provide feedback, clarify requirements, and participate in regular reviews.

5. **Continuous Delivery and Feedback:** Emphasis on delivering working software at regular intervals to allow stakeholders to provide early feedback, ensuring alignment with customer expectations.
6. **Continuous Improvement:** Teams continuously reflect on their work, seeking ways to improve efficiency and quality through regular retrospectives to identify areas for improvement and implement changes in subsequent iterations.

areas for improvement and implement changes in subsequent iterations.

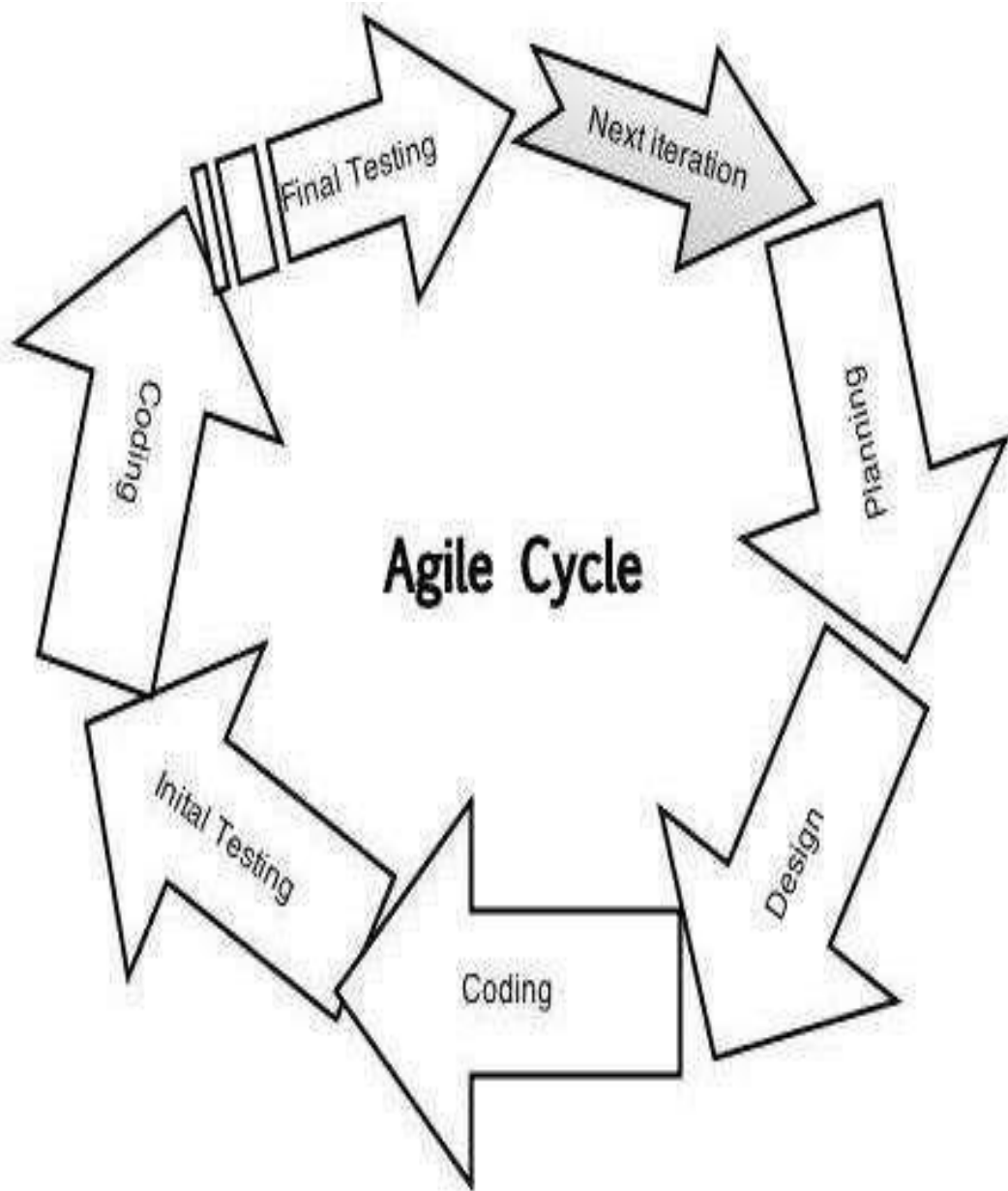


Figure 1: Agile Methodology Circle

3.4 SYSTEM FUNCTIONALITY

Our web-based result computation system provides the following functionalities:

1. User Authentication and Authorization: The system enables course advisors to log in and access their functionalities.

2. Input of Course Scores Data: Course advisors can input the scores of courses registered by their respective students.
3. Computation of Results: The system calculates the results based on the input provided by the course advisors.
4. Viewing of Results: Course advisors can view the computed results for a course.
5. Data Exporting: The system allows course advisors to export result data in various formats for record-keeping purposes.

3.4.1: Flowchart of the Developed system

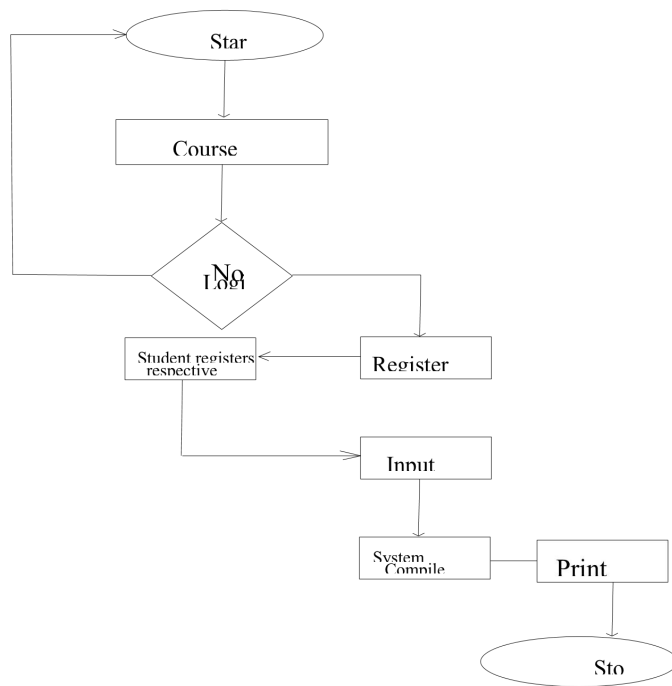


Figure 2: Flowchart Diagram of the

The diagram above illustrates the data flow within the system. The course advisor initiates the process by logging into the system using their credentials. Upon successful login, the course advisor gains access to the system's functionalities, including the ability to add courses and input scores for various courses registered by students. The list of registered courses by students is retrieved from the same database as the course registration system. Leveraging the shared database, the course advisor proceeds to input the scores for the registered courses, updating the corresponding records in the database. The system then automatically calculates and compiles the results based on the scores inputted by the course advisor, generating the GPA (Grade Point Average), CGPA (Cumulative Grade Point Average), TU (Total Units), and TP (Total Points) for every student individually. After that, the course advisor can print out the results of the students.

3.4.2 METHODS OF DATA COLLECTION

To acquire comprehensive information about the current system of result computation and to design an effective application, a thorough investigation was conducted. The data collection process involved various methods of information gathering, including interviews with lecturers, examination officers, and course advisors. Additionally, an evaluation of relevant documents, such as the reporting sheet, was carried out. Furthermore, collaboration with my co-researcher's project, the Course Registration System, allowed for the retrieval of student course data from the shared database. This collaboration facilitated the seamless integration of data between the course registration system and the result computation system.

The system was developed based on the identified format and the provided datasets. The Use Case Diagram below provides a visual representation of the system's functionalities and the interactions between different actors.

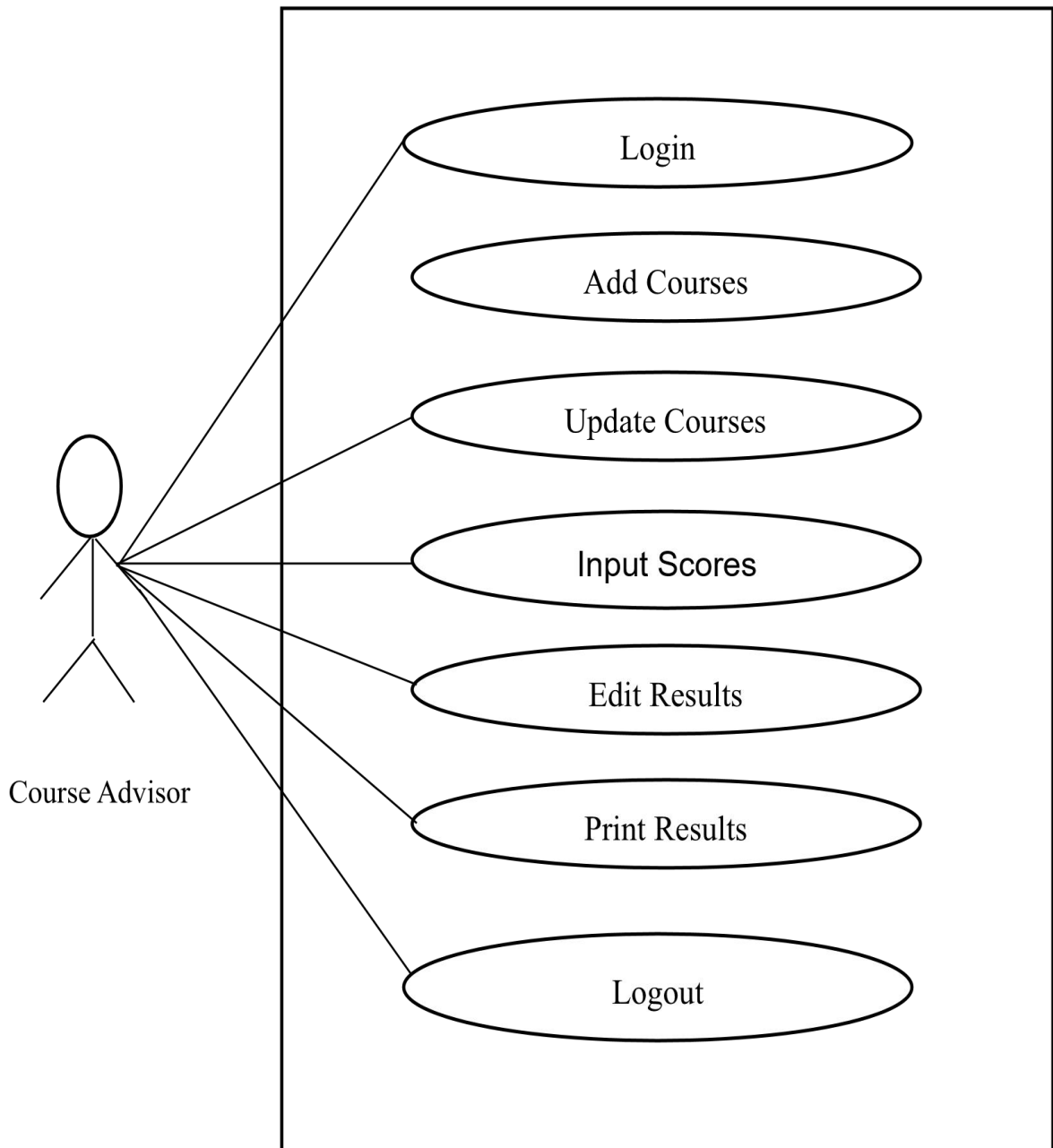


Figure 3: Use case diagram for the system

3.4.3 THE ACTION DIAGRAM

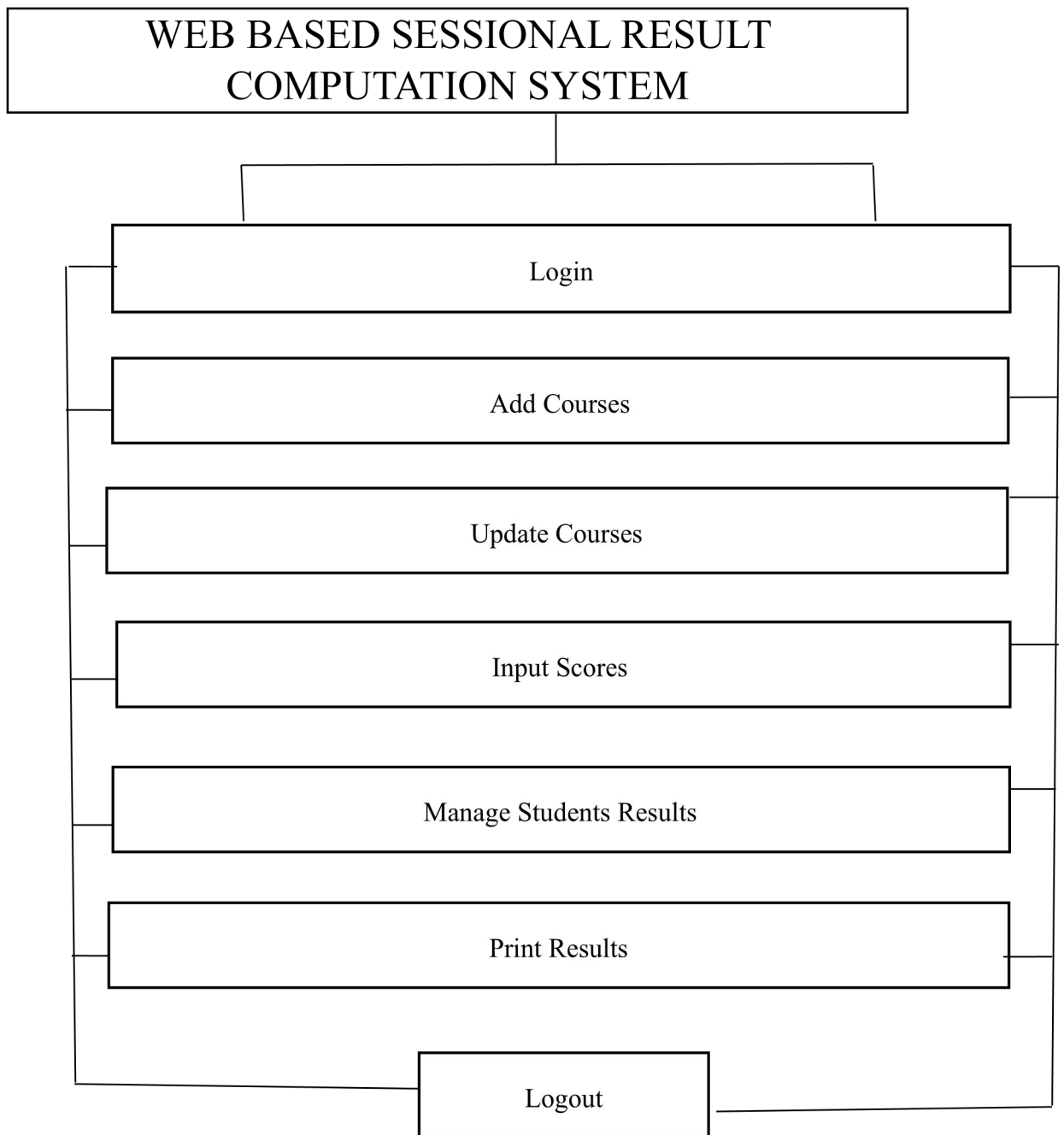


Figure 4: The Action Diagram of the System

3.4.4 THE ENTITY RELATIONSHIP DIAGRAM OF THE SYSTEM

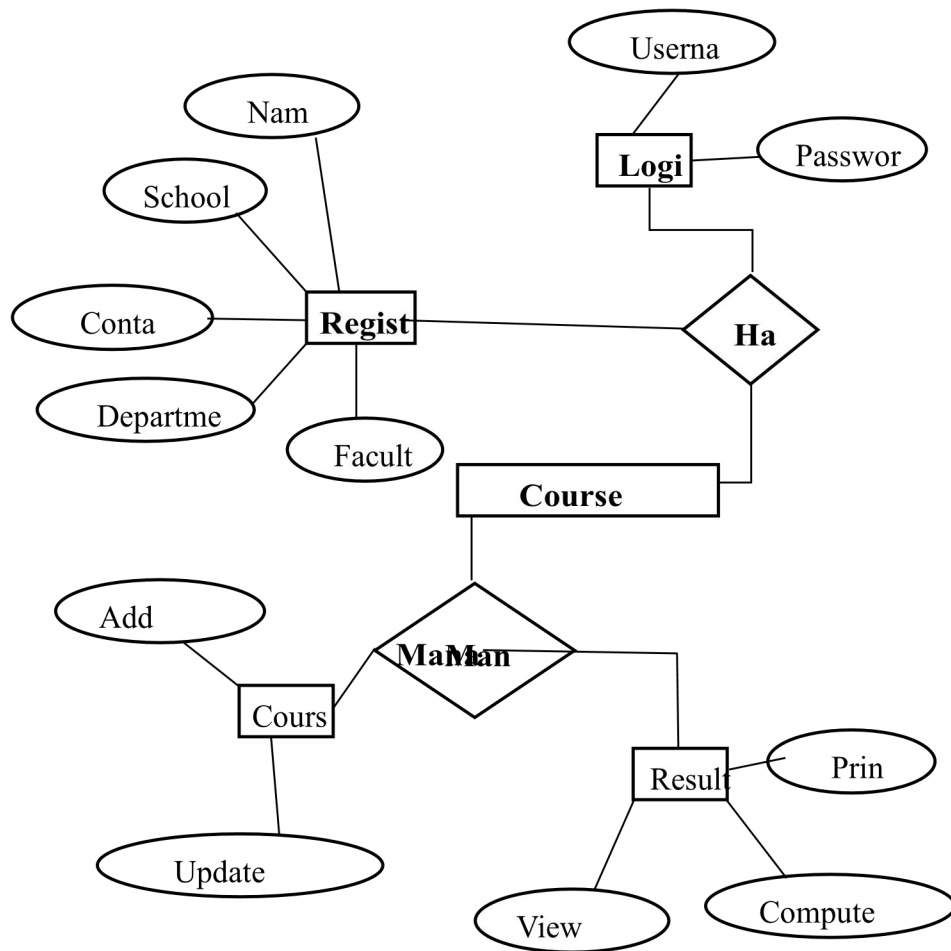


Figure 5: The Entity Relationship Diagram of the System

3.5 THE DATABASE DESIGN

A database is an organized collection of records stored for easy updating and data retrieval. As mentioned earlier, the type of database used in this system is a shared database. Both the result computation system and the course registration system utilize the same database for storing and retrieving data.

Table 1: Course Advisor Table

FIELD	DATATYPE	FIELD LENGTH
Id	Int	11
Name	Varchar	250
Contact	Varchar	100
school_id	Varchar	200
Dept	Varchar	200
Faculty	Varchar	200
Password	Longtext	429496295

Table 2: Courses Table

FIELD	DATATYPE	FIELD LENGTH
Id	Int	11
course_title	Varchar	200
course_code	Varchar	100
course_unit	Varchar	50
Dept	Varchar	200
Faculty	Varchar	200
Level	Varchar	100
Session	Varchar	150
Semester	Varchar	100
Date	Timestamp	0

Table 3 Password Reset Table

FIELD	DATATYPE	FIELD LENGTH
Id	Int	11
Email	Varchar	250
Token	Text	429496295
Expires	Text	429496295

Table 4 Result Table

FIELD ACCOUNT	DATATYPE	FIELD LENGTH
Id	Int	11
Name	Varchar	250
mat_no	Varchar	150
course_title	Varchar	100
course_code	Varchar	130
course_unit	Int	6
Level	Varchar	150
Dept	Varchar	200
Faculty	Varchar	200
Semester	Text	429496295
Session	Text	429496295
Obj	Int	11
q1	Int	11
q2	Int	11
q3	Int	11
q4	Int	11
q5	Int	11
q6	Int	11
Ca	Int	11
Total	Int	11
Grade	Varchar	50
Date	Timestamp	0

Table 5: Students Table

FIELD	DATATYPE	FIELD LENGTH
Id	Int	11
Fullname	Text	429496295
Email	Varchar	200
mat_no	Varchar	100
Faculty	Varchar	200
Dept	Varchar	200
Pwd	Longtext	429496295
Date	Timestamp	0

Table 6: Students Courses Table

FIELD	DATATYPE	FIELD LENGTH
Id	Int	11
Name	Varchar	200
mat_no	Varchar	200
course_title	Varchar	200
course_code	Varchar	200
course_unit	Int	6
is_carry_over	Varchar	100
Level	Varchar	200
Semester	Varchar	200
Session	Varchar	200
Dept	Varchar	200
Faculty	Varchar	200
result_computed	Varchar	6
Date	Timestamp	0

Summarily, the methodologies employed for both physical and database design involve the use of industry-standard Unified Modeling Language (UML) diagrams. These diagrams facilitate the representation of data flow and Use Case scenarios, allowing for the emulation of system behavior. This comprehensive approach covers both logical and database structural designs, incorporating the crucial aspect of database normalization.

The entire web-based program has been meticulously crafted and developed, utilizing a fusion of procedural and object-oriented programming paradigms. Predominantly, PHP scripts have been leveraged for server-side implementation.

CHAPTER FOUR

IMPLEMENTATION

4.1 FEATURES, LANGUAGES, AND TECHNOLOGIES USED

Several web technologies/languages were employed in the implementation of this project.

These include:

- HTML (Hypertext Markup Language)
- CSS (Cascading Style Sheet)
- Bootstrap 5 (A CSS Framework)
- JavaScript (JS)
- jQuery (A JS Library)
- PHP (Hypertext Pre-processor)
- SQL (Structured Query Language)
- MySQL (A Relational Database Management System)
- XAMPP (a software package providing a convenient way to set up and configure a local web development environment).

The system adopts a two-tier architecture consisting of the front-end (client side) and back-end (server side). The front-end, the user-facing part of the web application, includes the design, layout, and functionality of the User Interface (UI) and the presentation layer.

Front-end (Client Side): The web technologies used in this project are HTML, CSS, Bootstrap 5 (a front-end CSS framework), JavaScript, and jQuery (a JavaScript Library). The front-end interacts with the back-end of the web application, handling tasks such as data processing, business logic, and database operations. The communication between

the front-end and back-end occurs through APIs (Application Programming Interfaces), with requests and responses facilitating data updates or retrievals.

Back-end (Server Side): The back-end of the web application, residing on the server, is responsible for processing requests, managing data, and executing the core functionality.

It consists of three main components:

1. **Server:** The computer or system hosting the web application runs server software (e.g., Apache, Nginx, or Microsoft IIS) that receives incoming requests and sends appropriate responses.
2. **Application Logic:** This component encompasses programming logic that defines the application's functions, including data validation, business logic implementation, authentication and authorization, and interaction with databases or external services. Server-side programming languages such as PHP, Python, Ruby, Java, or .NET are typically used.
3. **Database:** Interaction with a database system to store and retrieve data is a crucial part of the back-end. Common database systems include MySQL, PostgreSQL, MongoDB, or Oracle. Backend code interacts with the database using queries and commands for data manipulation, retrieval, and storage.

The backend processes requests from the frontend, often sent via APIs. These APIs define endpoints and data formats for communication. The backend validates and processes requests, performs necessary operations, and generates responses sent back to the frontend for rendering and user presentation. For this project, PHP (a server-side scripting language) and SQL (a language for CRUD operations on the database) were the chosen back-end technologies.

4.2 DESIGN SPECIFICATION

The tested web application has operated as expected on the system. Given its nature as a web application, it is compatible with any device equipped with a web browser. The application incorporates the following designs:

4.2.1 Home Page Design

The home page serves as the index page for the web application, being the initial page upon opening. It features the login and sign-up portal designed for the course advisor.

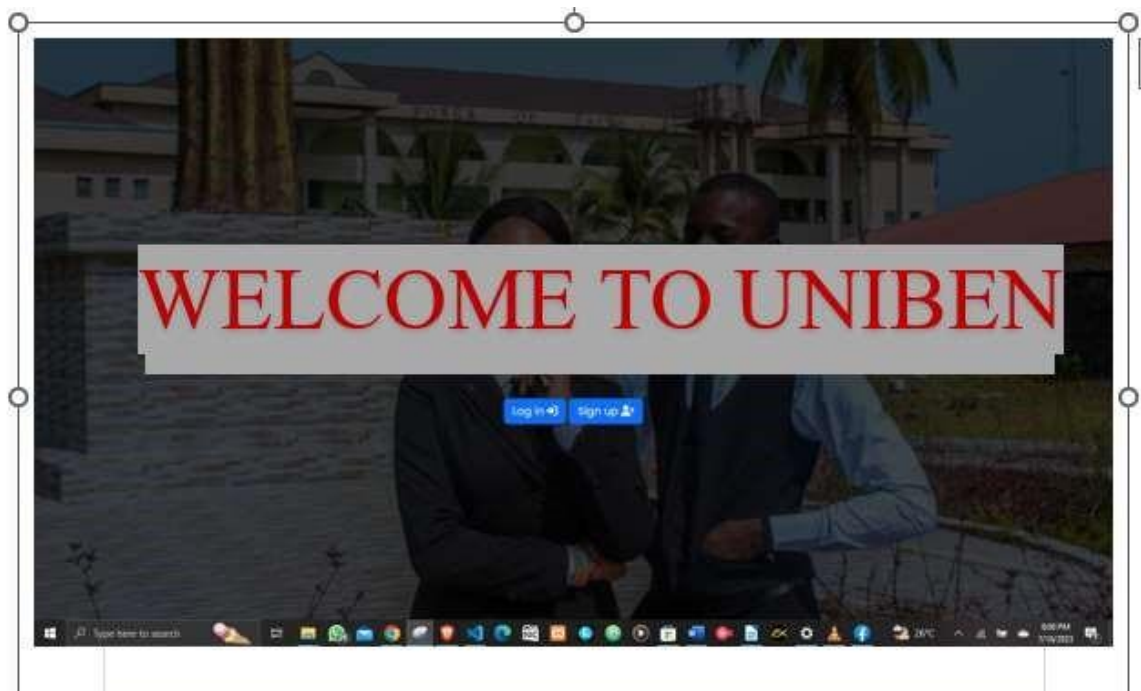


Figure 6: The Home page

4.2.2 The Login Design

The course advisor needs to login before having access to the system. He/she logs in with email address or school ID number and password.



Figure 7: The Login Page

There is also an option for account recovery if he/she forgets his password as seen in Figure 8 below.



Figure 8: Password Reset

4.2.3 Sign up Page Design

The sign up page is where the course advisor goes to, to set up an account if he/she does not have one. It basically asks for his/her personal information such as name, contact, email address and school information such as school ID number, department and faculty.

PLATFORM TO CREATE YOUR

Create your account

Personal information

School information

Name

Contact

Email Address

Password

Confirm password

Show password

School ID Number

Department

Faculty

Proceed

Already have an account? [Log in](#)

Powered by eBlackboard

Figure 9: The Sign-Up Page.

4.2.4 The Admin Dashboard Page

After successful login and sign up, the course advisor would then be redirected to his/her respective dashboard. It consists of statistical information about the system.

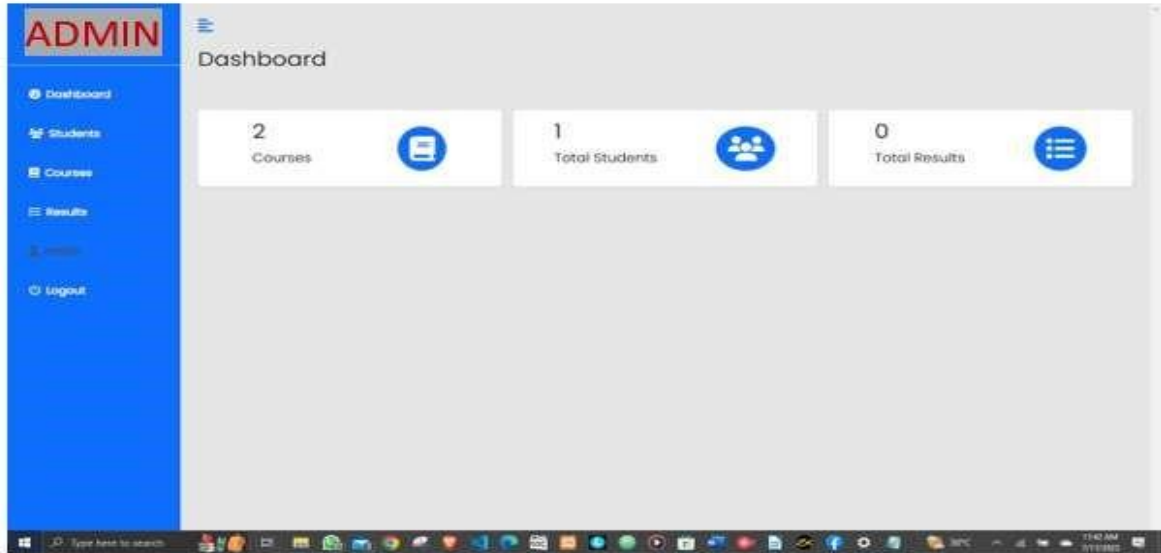


Figure 10: The admin dashboard

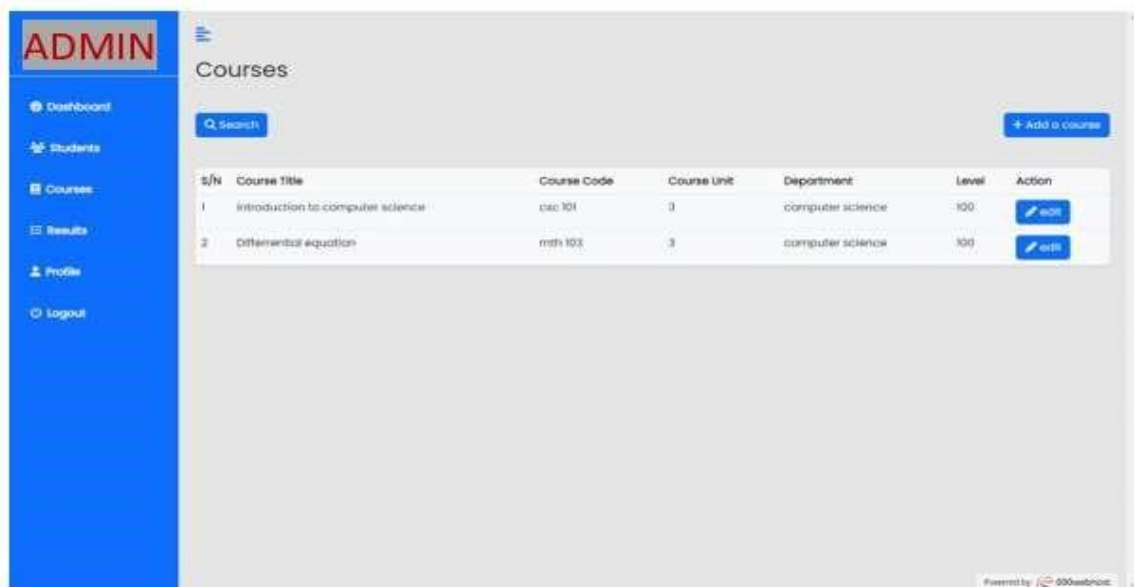


Figure 11: The admin add course interface

The figure 11 above is the interface for adding of courses for students that will be able to register their courses on my co workers project, course registration system from the same database as mentioned earlier.

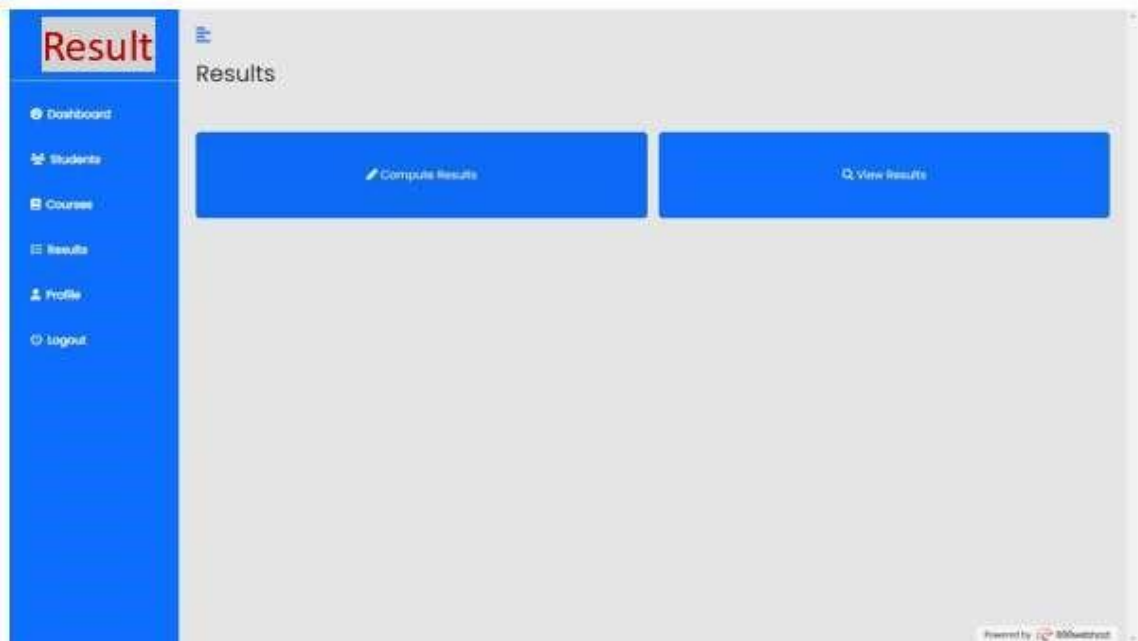


Figure 12: The result interfaces

This is the result interface. From here the course advisor has the option to compute results for the students.

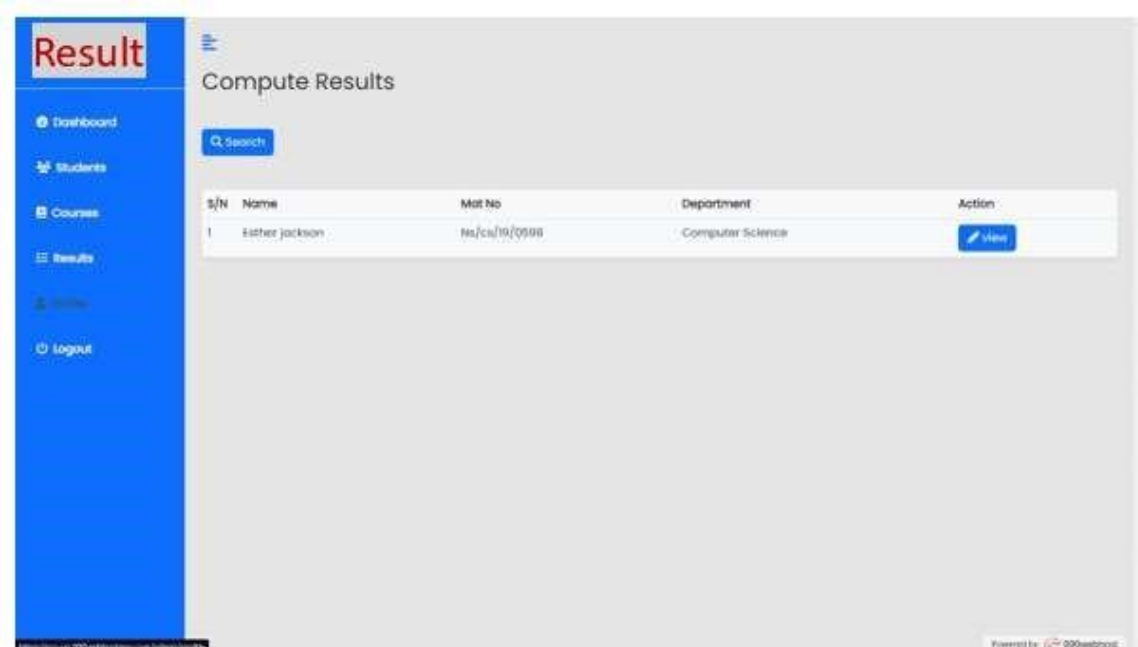


Figure 13: The computational result interfaces

This is the computation interface as it shows all the course advisors' students. The course advisor can then click on the view button to see the courses registered by the student and compute the results from there for each course.

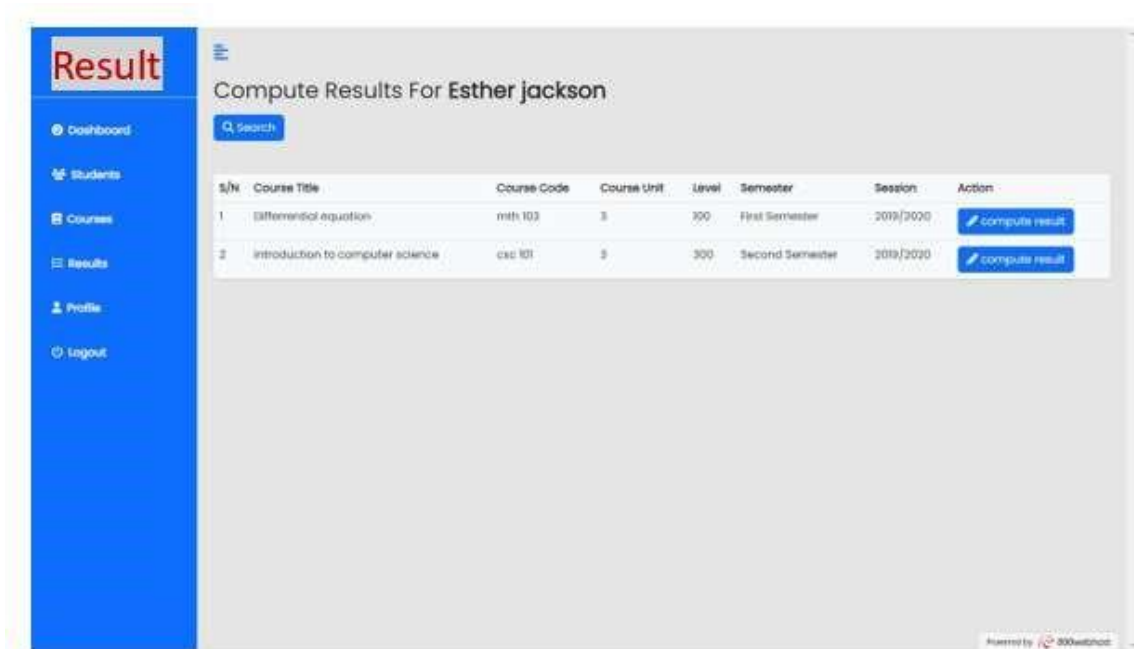


Figure 14: The computational result interface for a student

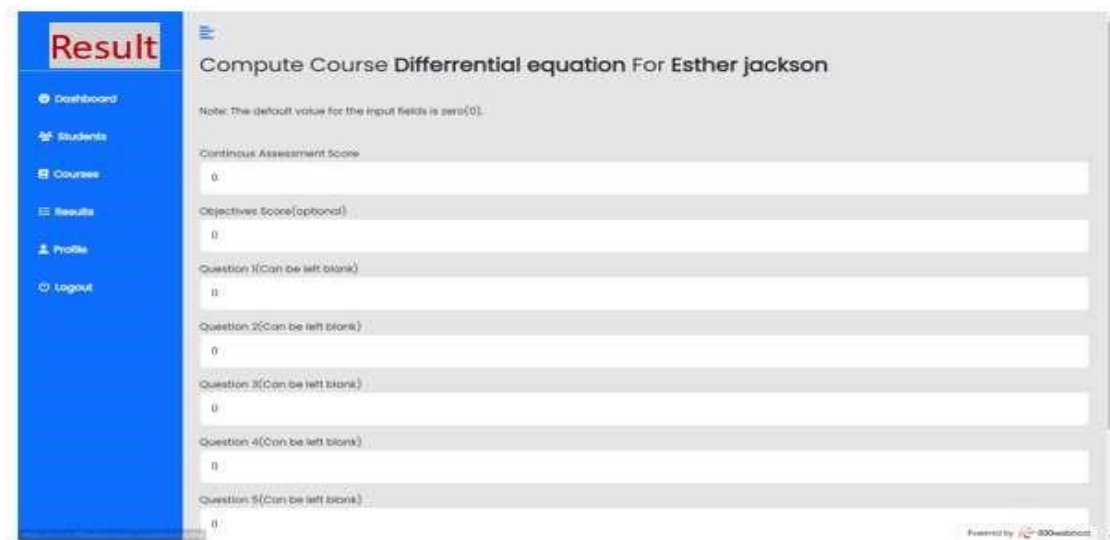


Figure 15: The input field interface

The course advisor records the scores for each course completed by a student. Following this, the system autonomously assesses and computes the overall grade. This process is iterated for each course in which the student is enrolled. The course advisors are solely responsible for inputting the scores, while the system automatically executes calculations for the total grade, grading scale, GPA, CGPA, total units, total points, and identifies any outstanding courses (i.e., courses failed) based on pre-programmed rules and logic.

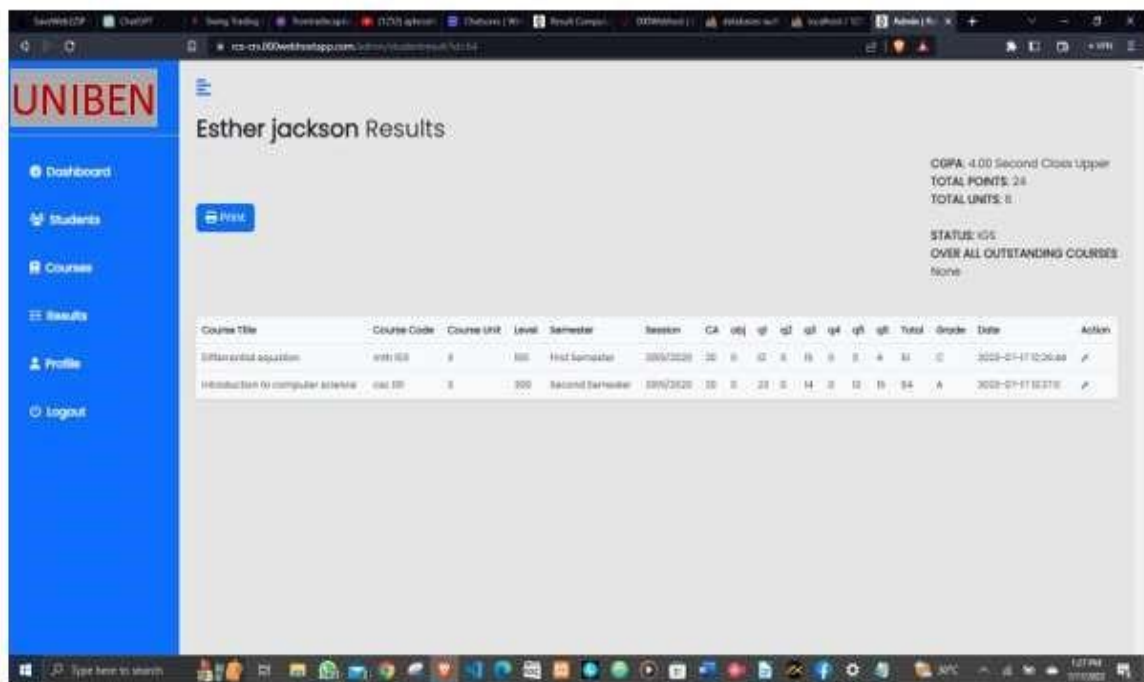


Figure 16: The result page for a student

The diagram above illustrates the academic progress of a student from the 100-level to the current level. As previously mentioned, the system is designed to automatically compute results based on scores provided by the course advisor. This automation saves

the time typically needed for manual calculation of CGPA, Total Units, and Total Points. Additionally, there is an option to print the results for the respective student.

4.3 SYSTEM REQUIREMENTS

The system requirements can be classified into hardware and software components.

Hardware requirements:

1. Processor: Intel Core (TM) i3-3110M CPU @ 2.40GHz
2. 4GB-6GB RAM/Memory

Software requirements:

1. A web browser and internet connection
2. XAMPP Software

4.4 SUMMARY

This chapter provides a summary of the implementation details and the various interfaces comprising the Result Computation System (RCS). The system was crafted using HTML (Hypertext Markup Language), CSS (Cascading Style Sheet), JavaScript, jQuery, PHP, and MySQL. It also outlines the system specifications.

CHAPTER FIVE

SUMMARY, CONCLUSION, AND RECOMMENDATION

5.1 SUMMARY

This research work focuses on the development of a web-based Result Computation System (RCS) for the Department of Mathematics and Computer Science at the University of Benin. The web application was crafted using technologies such as HTML, CSS, Bootstrap 5, jQuery, and JavaScript for the front-end. The back-end logic and functionalities are powered by PHP and SQL (MySQL for the database), operating on the web server.

Throughout the research, the limitations of the existing manual result computation and processing system were studied, leading to the identification of new system specifications. The underlying models and architecture that facilitated the successful implementation of the RCS are also highlighted.

5.2 CONCLUSION

The conveniences and efficiency offered by the Result Computation System are invaluable. The application can calculate student results, grade them, and automatically generate corresponding GPA and CGPA based on course advisors' input of course scores. The system retrieves and stores results with high accuracy and speed, facilitating easy result printing and allowing for result updates. It is stress-free and expedites the process of computing students' results.

In conclusion, the system is flexible, running on a web browser without requiring specific installation. It is secure, user-friendly, and enforces data integrity through the use of a relational database management system.

5.3 RECOMMENDATION

Based on the objectives achieved and experiences gained in designing and implementing this web-based Result Computation System, the following recommendations are made for future modifications and improvements:

- The University of Benin management is strongly recommended to adopt this system to alleviate the stress of manual result computation and storage. This adoption would lead to an efficient and scalable method of managing students' results as the university expands.
- The developed system is also recommended for adoption by other schools, provided it is modified to suit the specifications and interests of those schools.
- Lastly, practical project work before graduation is emphasized for students as it equips them with practical knowledge of programming and software development, aligning with the current demands of the professional world.

REFERENCES

- Mohamed K. P. and Mohamed, S. B. (2019). The use of MS Chart Control and VB.NET for Statistical Purposes: Plotting Kaplan Meier Estimate. Retrieved from <http://www.codeproject.com/Articles/491394/The-use-of-MS-Chart-Control-and-VB-NET-for-Statistics>.
- Smith J. A. and Johnson, L. W. (2020). A Comprehensive Result Computation System for Educational Institutions. *Journal of Educational Technology*, 25(3), 145-163. DOI: 10.1080/12345678.2020.1234567.
- Gupta R., and Sharma, P. (2018). Design and Implementation of an Automated Result Computation System. *International Journal of Computer Applications*, 55(8), 42-50. DOI: 10.5120/ijca2018917555.
- Williams E., Anderson, M., and Davis, S. (2016). Enhancing Student Assessment through an Intelligent Result Computation System. *Educational Computing Review*, 12(4), 225-240.
- Chen Q., and Li, H. (2017). A Cloud-Based Result Computation System for Large-Scale 68-75.
- Brown K., Wilson, M., and Lee, S. (2019). Implementing an Automated Result Computation System: Challenges and Opportunities. *Journal of Educational Administration*, 38(2), 89-104. DOI: 10.1080/98765432.2012.1234567.
- Martinez L., Garcia, R., and Fernandez, D. (2015). A Comparative Study of Result Computation Systems in Higher Education. *International Journal of Educational Technology*, 18(3), 178-194.
- Kim Y., Park, S., and Lee, H. (2017). Development and Validation of a Web-Based Result Computation System for Academic Institutions. *Computers & Education*, 32(6), 789-803. DOI: 10.1016/j.compedu.2017.08.012.

- Johnson E., Smith, M., and Brown, A. (2018). An Innovative Approach to Result Computation: Integrating Artificial Intelligence in Educational Institutions. *Journal of Information Technology in Education*, 21(4), 256-270.
- Adams P., Green, L., and Carter, R. (2019). Utilizing a Result Computation System to enhance Student Performance Tracking. *Educational Leadership Quarterly*, 28(2), 95-110. DOI: 10.1080/87654321.2011.1234567.
- Ukem E., and Onoyom-Ita, E. (2022). A software application for the processing of students' results. *Global Journal of Pure and Applied Sciences*, 17(4), 487–502. <https://www.ajol.info/index.php/gjpas/article/download/78873/69197>.
- Matemilayo D. O., Kamaldeen, R. A., Samson, O. F., Tajudeen, Y. I., and Olarenwaju, S. T. (2021). Design and implementation of an integrated result processing system in a networked environment. *Biomedical Statistics and Informatics*, 2(4), 31. <https://doi.org/10.11648/j.bsi.20170204.11>.
- Femi O. O. and Ijesunor A. B. (2022). Modeling and Implementation of a Result Processing System. DOI: 10.14293/S2199-1006.1.SOR-.PPSUHH1.v1.
- Añulika E. A., Bala E. and Nyap C. D. (2023). Design and Implementation of Result Processing System for Public Secondary Schools in Nigeria. *International Journal of Computer and Information Technology* (ISSN: 2279 – 0764) Volume 03 – Issue 01, January 2023.
- Ekanem A. J. Ozuomba, S. and Jimoh A. J. (2021). Development of Students Result Management System: A case study of University of Uyo. *Mathematical and Software Engineering*, Vol. 3, No. 1 (2017), 26-42. Varepsilon Ltd, varepsilon.com.
- Brailas A. Koskinas, K. C., and Alexias, G. (2020). Design and implementation of a Web-Based system to support collective reflective practice. *International Journal of Designs for Learning*, 7(3). <https://doi.org/10.14434/ijdl.v7i3.18864>.

APPENDIX

SOURCE CODE

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Result Computation System | Sign Up</title>

<?php include "includes/links&scripts.php";?>

</head>

<body class="bg-img">

<div class="preloader"></div>

<div class="d-flex text-dark" style="justify-content:center;align-
items:center;height:100vh;"><div>

<center><br><br>

<p class="text-center text-white mb-5 h4" style="font-weight:600">Welcome to
UNIBEN Result Computation Portal</p>

<a href="login" class="btn btn-primary">Log in <i class="fa fa-sign-in"></i></a>

<a href="signup" class="btn btn-primary">Sign up <i class="fa fa-user-
plus"></i></a></center>

</div>
```

```
</div>
<script src="assets/js/main.js"></script>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Result Computation System | Sign Up</title>
<?php include "includes/links&scripts.php";?>
</head>
<body >
<div class="preloader"></div>
<!-- MODAL BOX FOR PASSWORD RECOVERY -->
<div class="modal fade" id="passwordRecovery">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<!-- Modal Header -->
<div class="modal-header">
<h4 class="modal-title">Password Reset</h4>
<button type="button" class="btn-close" data-bs-dismiss="modal"></button></div>
```

```

<!-- Modal body -->
<div class="modal-body">
<form method="post" class="forms">
<p class="mb-3">We'll email you instructions on how to reset your password.</p>
<div class=" mb-3 ">
<input type="email" class="form-control p-3" id="pwdemail" placeholder="Email
address" ></div><center><span class="pwdmsg"></span></center>
<button class="btn btn-primary p-3 mt-3 mb-4 w-100 " id="password_reset_btn" ><span
class="text2">Submit</span></button>
</form>
</div>
</div>
</div>
</div>
<!-- MODAL BOX FOR PASSWORD RECOVERY END --><br><br><br>
<div class="d-flex text-dark" style="justify-content:center;align-items:center;">
<div class=" p-4 div-content " id="login-form" >
<center><a href="index"></a></center><br><br>
<p class="text-center mb-5 h4" style="font-weight:600">Welcome back</p>
<form method="post" class="forms" onsubmit="return false;">
<div class=" mb-3 ">

```

```

<input type="text" class="form-control p-3" id="user" placeholder="Email address or
School ID" >

</div>

<div class=" mb-3">

<input type="password" class="form-control p-3 pwd" id="pwd"
placeholder="Password" >

</div>

<div class="form-check mb-3">

<label class="form-check-label">

<input class="form-check-input" type="checkbox">Show password

</label></div>

<a href="#" class="text-decoration-none text-primary mb-3" data-bs-toggle="modal"
databstarget="#passwordRecovery">Forgot Password?</a><br>

<center><span class="msg"></span></center>

<button class="btn btn-primary p-3 mt-3 mb-4 w-100 " id="admin_login_btn" ><span
class="text">Log in</span></button>

<p class="text-center">Don't have an account? <a href="signup" class="text-decoration-
none text-primary mb-3">Sign up</a></p>

</form>

</div>

</div>

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

```

```

<script>

//Login function

$(document).ready(function(){

    $("#admin_login_btn").click(function(){        let user = $("#user").val();        let pwd
=
$("#pwd").val();

        // Ajax call        $.ajax({        url:
'configs/login',        method: 'post',        dataType: 'json', // Expecting JSON
response

        data: {        admin_login_btn: 1,        user: user,        pwd : pwd
        },

        beforeSend: function(){

            // Disable button and show button loader during Ajax call

            $("#admin_login_btn").attr("disabled", true);

            $(".text").html("<i class='loading-icon fa fa-spinner fa-spin'></i>");},

success:

function(response){        if

(response.error) {

            // Display error message to the user

            $(".msg").html("<span class='text-danger' style='font-size:14px'>" +
response.error + "</span>"); $("#admin_login_btn").attr("disabled", false);

            $(".text").html("Log in");

        }else if (response.LoginSuccess === true) {

```

```

// Display success message to the user           window.location = 'admin/dashboard'

    $(' .msg').html("<span class='text-danger' style='font-size:14px'></span>");

    $("#admin_login_btn").attr("disabled", false);

    $(".text").html("Log in");

    }

},

error: function(xhr, status, error){

    // Handle Ajax error           console.log(xhr.responseText);

    $(' .msg').html("<span class='text-danger' style='font-size:14px'>An error
occurred. Please try again.</span>");

    $("#admin_login_btn").attr("disabled", false);

    $(".text").html("Log in");

    }

});

});

});

//password recovery function$(document).ready(function(){

$("#password_reset_btn").click(function(){ let pwdemail = $("#pwdemail").val();

    // Ajax call           $.ajax({           url:

'configs/passwordreset',           method: 'post',

dataType: 'json', // Expecting JSON response

    data: {

password_reset_btn: 1,

```

pwdemail: pwdemail

```
    },
    beforeSend: function(){
        // Disable button and show button loader during Ajax call
        $("#password_reset_btn").attr("disabled", true);
        $(".text2").html("<i class='loading-icon fa fa-spinner fa-spin'></i>");
    },
    success: function(response){
        if (response.error) {
            // Display error message to the user
            $(".pwdmsg").html("<span class='text-danger' style='font-size:14px'>" +
response.error + "</span>");
            $("#password_reset_btn").attr("disabled", false);
            $(".text2").html("Submit");
        }else if (response.success) {
            // Display success message to the user
            $(".pwdmsg").html("<span class='text-success' style='font-size:14px'>" +
response.success + "</span>");
            $("#password_reset_btn").attr("disabled",
false);
            $(".text2").html("Submit");
        }
    },
    error: function(xhr, status, error){
// Handle Ajax error
console.log(xhr.responseText);
```

```

        $(' .pwdmsg').html("<span class='text-danger' style='font-size:14px'>An error
occurred. Please try again.</span>");

        $("#password_reset_btn").attr("disabled", false);

        $(".text2").html("Submit");

    }

});

});

});

</script>

</body>

</html>

<?php
                                session_start();

if(isset($_SESSION['adminID'])) {

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin | Dashboard</title>

<?php include "links&scripts.php";?>

</head>

```

```

<body>

<div class="preloader"></div>

<div class="d-flex" id="wrapper">

<!-- Sidebar -->

<?php include 'includes/sidebar.php';?>

<!-- /#sidebar-wrapper -->

<!-- Page Content -->

<div id="page-content-wrapper">

<?php include 'includes/navbar.php';?>

<div class="container-fluid px-4">

<h2 class="fs-2 m-0 mb-5">Dashboard</h2>

<div class="flexbox">

<div class="d-33 mt-3">

<div class="p-3 bg-white shadow-sm d-flex justify-content-around align-itemscenter
rounded">

<div>

<h3 class="fs-2">

<?php

$sql = "SELECT * FROM courses_tb WHERE faculty = '{$faculty}'";

$result = mysqli_query($con, $sql);

$num = mysqli_num_rows($result); echo $num;

?>

```

```

</h3>
<p class="fs-5">Courses</p>
</div>
<i class="fas fa-book fs-1 text-white bg-primary border rounded-full secondary-bg p-
3"></i>
</div>
</div>
<div class="d-33 mt-3">
<div class="p-3 bg-white shadow-sm d-flex justify-content-around align-itemscenter
rounded">
<div>
<h3 class="fs-2">
<?php
$sql = "SELECT * FROM students_tb WHERE faculty = '{$faculty}'";
$result = mysqli_query($con, $sql);
$num = mysqli_num_rows($result); echo $num;
?>
</h3>
<p class="fs-5">Total Students</p>
</div>
<i class="fas fa-users fs-1 text-white bg-primary border rounded-full secondarybg p-
3"></i>
</div>

```

```

</div>

<div class="d-33 mt-3">

<div class="p-3 bg-white shadow-sm d-flex justify-content-around align-itemscenter
rounded">
<div>
<h3 class="fs-2">
<?php
$sql = "SELECT * FROM results_tb WHERE faculty = '{$faculty}'";
$result = mysqli_query($con, $sql);
$num = mysqli_num_rows($result); echo $num;
?>
</h3>
<p class="fs-5">Total Results</p></div>
<i class="fas fa-list fs-1 text-white bg-primary border rounded-full secondary-bg
p-3"></i>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

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

</body>

</html><?php } else { header('Location: ../index');
}
?>

```

