CHAPTER ONE

INTRODUCTION

This chapter goes through the project description as well as some background information. Furthermore, this chapter follows the problem description of the project to offer a clear knowledge of the project's scope and objectives. This chapter will act as a guideline for all later stages of development.

1.1   Background of The Study

As a result of technological advancements and the rapid pace of globalization, higher education institutions are now either partially or entirely involved in an online environment. The recent expansion of computer networks, specifically the World Wide Web (WWW), has enabled individuals to communicate, socialize, and interact via the internet.

Completing a final year project is part of the criteria required to graduate as a higher national degree holder from a polytechnic; it is difficult since students will apply their skills to construct real-world transforming applications.

Allocation of students to supervisors, assessment procedures, access to resources, assistance and supervision, and overall project management are all important factors that influence students' learning experiences throughout their final year project. Hussain, et al, 2020, as cited in Teo, 1998

An assessment may be an extremely important component in the teaching and learning environment since it should progress knowledge as well as quantify or verify results. E-assessment is the use of technology to organize and provide assessments that might be diagnostic, summative, or formative in nature. Huda, S. & Siddiq, T. (2020).

Since the inception of the Computer Science department, assigning students to supervisors and assessing student projects and seminars is done manually. Assessment of students is done in two ways, the student supervisor would assess the student as well as the panel which would access the student twice; namely the seminar and final defense assessment. This is done to ensure that the student carries out their project themselves or participated, which will then be used to grade them. The assessment sheet has to be printed in copies by the student depending on the number of assessors in that particular venue, similarly, with assigning students to supervisors, considering the number of students is not easy to assign and group students

probably using excel, it is monotonous and quite inefficient. This approach has been taken repeatedly over the years and causes a lot of paperwork with the possibility of poor documentation. This study will develop a student-to-supervisor allocation and assessment system that will be used by the department in assessing student project and seminar defense, thereby increasing work efficiency among project coordinators and defense panelists. The system uses an algorithm to automatically assign students to a supervisor, which creates an avenue where students can easily find the supervisor's contact information for ease of communication.

1.2   Statement of The Problem

It has been a mere saying that students are to be allocated to supervisor on time but the reverse is the case due to the series of monotonous activities involved in the process, not enough time is given to the student in completing the project, even when allocated student find it difficult in reaching out to the supervisors, a student would deliberately refuse to come for the defense until the last day increasing workload for the assessors. A lot of paperwork is involved in assessing students both in the seminar and project defense, which if not well documented may lead to incorrect grade input for that particular student, the highlighted problems jogged my interest to embark on the project.

1.3   Aim and Objectives of the Study

To develop a student-to-supervisor allocation and assessment system for the computer science department at Kaduna polytechnic.

Objectives

The objectives of this research work are as follows:

i.   Student and supervisor data sets will be extracted from the department depending on the student that qualified for that year's project as both the student and supervisors are not to perform registration on the site. The registration is automated

ii. Django which is a high-level python web framework will be employed in the backend development, Modern technologies like HTML, CSS, and JavaScript will be employed in front-end development, and open-source relational database; MySQL will be employed as the database technology.

1.4   Scope of the Study

This research work is centered on the development of an algorithm for assigning students to supervisors and assessment of student seminars and project defense, as well as the development of a user-friendly, web-based interface to serve its purpose and implementation of the system and evaluation of its effectiveness. It will not cover the overall functionality of a project management system, including submitting project topics for approval and others.

1.5   Limitations of the Study

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

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

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

Finance – Some material that would have facilitated the research was restricted to payment before access which prevented the researcher to have an in-depth study and analysis of the subject matter.

1.6   Significance of Study

This study would increase the drive for improvement in such a way that it will provide an avenue for other parts of a project management system to be included. It is carried out to eliminate time wastage in allocating students to their various supervisors, thereby providing a secure way for the higher institution of learning to create and manage the data related to student-supervisor relationships, as well as providing a platform where assessors can assess student project and seminar defense and manage the data efficiently.

1.7   Project Organization

The project is divided into five chapters. The outlines are presented below:

Chapter One: Introduction

Chapter one introduces this project work, the background of the study, the statement of the problem, the aim and objectives, the scope of the study, limitations of the study, the significance of the study, project organization, and the definition of terms.

Chapter Two: Literature Review

This chapter focuses on the literature review, and the contributions of other scholars on the subject matter being discussed.

Chapter Three: Methodology and Design

This chapter is concerned with the presentation of the results of system analysis and design. It presents the research methodology used in the development of the system to facilitate an understanding and effective future implementation of the system.

Chapter Four: System Implementation Evaluation

This chapter describes the system implementation and documentation, analysis of modules, and system requirements for implementation.

Chapter Five: Summary, Conclusion, and Recommendation

The chapter provides a summary of major findings, conclusions, and recommendations based on the study conducted.

1.8   Definition of Terms

i.   **Project:** A project is a collection of activities that must be accomplished over a set period while keeping cost and other constraints in mind to achieve a specified conclusion or objective.

ii.   **Project Allocation**: this is the process of assigning a specific project to a student and a supervisor. The supervisor will be responsible for providing guidance, feedback, and support to the student throughout the project and will also be responsible for assessing the student's work.

iii. **Project Assessment**: this is a process of evaluating a student's work on a project over some time. It involves assessing the quality of the project, its progress, and the student's understanding of the project's scope.

iv.   **Algorithm**: An algorithm is a set of instructions or steps used to solve a problem or accomplish a task. Algorithms are used in computer programming to describe a set of operations that need to be performed for a computer to complete a certain task.

v.   **Django**: Django is an open-source web framework written in Python. It is designed to help developers create complex, database-driven websites with minimal coding.

vi.   **System**: A system is a set of elements or components that are organized for a common purpose. It can refer to a set of components that interact to form a complex whole, or to a set of procedures or rules that are followed to achieve a specific goal.

vii. **Database**: A database is a system intended to easily organize, store, and retrieve large amounts of data. It consists of an organized collection of data for one or more uses, typically in digital form.

CHAPTER TWO

LITERATURE REVIEW

2.1   Introduction

This chapter aims to explain how the topic under research relates to prior research, current practice, or other areas of knowledge by mentioning relevant works by other scholars that have dealt with a related issue. Furthermore, this chapter will present a synthesis of the current research, highlighting areas of agreement, disagreement, and gaps in the literature, to show the relevance of the project topic in the field.

2.2   Literature Review

Ojo et al. (2022). Design and implementation of student project management and allocation system. The use of a manual system in the administration and assignment of projects to students causes several issues such as duplication of approved topics, inappropriate recording of assigned project topics, and so on. These issues spurred the researcher's development of a student project management and allocation system to address them.

Interviews were done to acquire more information to aid in the creation and design of the system. The proposed concept is a two-tier architecture with a backend and a front end. The development technologies employed in the development of this research include HTML, CSS, JavaScript, PHP, MySQL, and XAMPP server are the programming languages and tools used. This language was ideal for constructing internet applications due to its object-oriented capabilities and class buries.

This proposed web-based system handles the activities of the "student project management and allocation", it handles the database and keeps track of all students or groups of students who have registered as finalists on the site, as well as those students who have been shortlisted and meets the eligibility requirements provided by the lecturer/supervisor. The downside of the system is that finalists have to register on the site themselves which implies that anyone can register on the site and the allocation process is not automated.

Amadi et al. (2021). Design and Implementation of Students' projects allocation system. The use of a manual method in project administration and delivery presents several problems, such as project supervisors not knowing whether a given topic has previously been authorized for a student, inefficiency in entering, updating, and retrieving project records, difficulty obtaining

softcopies of previous projects, duplication of authorized themes, improper documentation of allocated project topics and so on.

Furthermore, in the construction of this system, the researchers used the widely recognized software engineering paradigm - Model View Controller (MVC). This research technique involves the procedure for data gathering and analysis required to identify or address the topic for which the study is being conducted. The data acquired for this study project allowed the researcher to gain an understanding of how the present system operates and also aided in the design of the new system. Data was gathered using both primary and secondary sources which include interviews with students and faculty, observations, and document study. MySQL was utilized as the database, PHP was used for the backend development, and HTML, CSS, and, JavaScript were employed for the frontend development.

In conclusion, the developed project allocation system allows the finalist to apply for projects online, which the supervisors will mark, and the system will allocate a project to the student based on the highest score awarded by the supervisor to the student's proposals. The adoption of a web-based technique has several advantages, including the ability to access information from anywhere and at any time of day and it will close the communication gap between students and their supervisors, resulting in improved service delivery. Due to the developmental technologies used, the system might lack scalability if there is a high influx of users.

Adamu A. (2020). Final Year Student Project Allocation Archiving and Management System. The study described numerous ways for allocating, archiving, and managing projects, emphasizing that the approaches are time-consuming, stressful, inconvenient, and incur large costs for typing and printing documents that will be presented to the supervisor. Another important issue is student duplication of project subjects, which occurs when two supervisors accept the same topic for two or more students in the same department as well as the mishandling of writeup. As a result, a user-friendly, effective, efficient, and convenient solution is required to address the aforementioned challenges.

Moreso, the data collection was made in form of a questionnaire which is a primary source of data collection. The system is built utilizing the PHP programming language, ASP.NET to provide the Graphical User Interface (GUI), and MySQL paired with XAMPP for the database. The system is intended to work with the Windows operating system.

Furthermore, a well-structured questionnaire was administered to assess the usability and convenience features of the manual method and the proposed system. According to usability findings, 89% of respondents believed that the designed system is easier to use than the old technique. The results of convenience evaluations indicated that 95% of respondents agreed that the created system provides a more convenient service to both students and supervisors than the old way.

In conclusion, the system's functionality demonstrates that it functions properly. The technology may be used to replace the manual approach of overseeing final-year students at any higher education institution. It will lessen the difficulties, energy, and time needed to monitor and manage the final-year student project. The study tackles the above-stated problem by establishing a final-year student project allocation, archiving, and management system in which the supervisor and student may engage in real-time checkmate for project approval and submission. The downside of the system is that it poses a security threat as users' information is stored in its session as long as possible unless they manually log out.

Huda et al. (2020). E-Assessment in Higher Education: Student's Perspective. There are various traditional assessment methodologies used in higher education institutions to measure academic progress, such as paper-pencil tests, presentations, assignments, and many more. The study attempts to examine the benefits that students receive from e-assessment as well as the challenges that they face when institutions adopt e-assessment and move from conventional evaluation.

Furthermore, the data was gathered from both primary and secondary sources. The core data was gathered through a survey of 200 Bangladeshi undergraduate and postgraduate students drawn at random from Dhaka's institutions. The survey instrument was made up of 27 statements on a Likert scale that were created using information from the literature. Secondary data sources included journals, research papers, and websites.

In conclusion, this study explores the usefulness of e-assessment in higher education from the perspective of students, as well as the student's attitude to this technique. The new assessment method helps both institutions and students. The report is based on the opinions of a small sample of university students picked at random, who had varied reactions to e-assessment. Although students recognize the value of e-assessment, they are concerned about technology-

based exams since not all of them have the same degree of IT ability. More studies should be conducted to investigate additional issues of e-assessment in higher education.

Martin and Fanus (2018). E-Assessment in Higher Education: A Review. Most universities that use conventional assessment in the form of high-stakes examinations encounter a variety of malpractices, such as professors accepting payments to leak questions or invigilators favouring certain students. Other issues confronting traditional assessments include the burden placed on lecturers in terms of marking, organizing, and recording student scripts, the costs associated with printing examination papers or assessments, security concerns, and, as student numbers in higher education institutions increase is also an issue.

Moreso, the advancement of technology and e-learning systems has resulted in an increased demand for methods and means of assessing students in such a system. Assessment is an essential component of every higher education institution's teaching and learning process. The purpose of this research is to give a discussion on e-assessment that focuses on ideas such as e-assessment definitions, e-assessment delivery platforms, tasks that may be obtained through e-assessment, benefits and problems of e-assessment, and e-assessment principles.

It is concluded that e-assessment can be beneficial provided the assessment is credible and lecturers work together to develop authentic, consistent, transparent, and practicable assessments. E-assessment can also be used to assess higher-order assessment tasks.

Supianto and Khaerudin. (2020). Web-based project assessment. This is a library research project that aims to explore the literature linked to research difficulties by selecting, reading, studying, and researching research relevant to the topic of this project. The researcher briefly defines web-based assessment and then maps the benefits of web-based evaluation.

Furthermore, data was gathered from literature reviews on emotional evaluations and research on the use of social media in learning and assessment. Editing, categorization, and interpretation were used to analyse the data.

In conclusion, web-based assessment refers to assessment models such as e-tests, e-assessments, computer-based tests, and internet-based tests. Web-based assessment is more flexible, reduces paper consumption, data collecting, and analysis quickly gives immediate feedback, simplifies instructor assignments, and encourages e-learning. However, there are other roadblocks, including the aesthetic style of the online interface, which may cause issues

for consumers. Users lose too much time due to ineffective menu layout and bad navigation. It is also vital to guarantee that students can efficiently use the online learning system. In reality, web-based project appraisal can modify current assessment methodologies such as peer evaluation, self-evaluation, and group evaluation. These are alternate exams that educators might use to evaluate student efforts on the web.

Arumugam et al. (2021). Academic Project Information Management System. Many universities now manage final year project data in an offline fashion, which includes spreadsheet entries for all groups, manual group formation and supervisor assignments, and retaining paper copies of the materials given by students. When there are more groups, there is a greater chance of errors occurring during the updating process.

Moreso, the article describes how a web-based automated system would resolve all faults and mistakes while remaining operational offline. The goal of this research is to offer a system for managing groups, automatic guide allocation, document sharing, smoothing the process of communication between guides and students, keeping a log of all actions, and monitoring student project progress.

Hassan (2016). A preemptive goal programming for allocating students into academic departments of faculty. A goal programming model is created to maximize the placement of students in a faculty's academic departments. The goal programming approach considers space capacity, budget allocation, the number of teachers, and affirmative action quotas as goal constraints that must be met. Each constraint has a priority level and a weight associated with it. This goal programming paradigm is then applied to the University Kebangsaan Malaysia's Faculty of Science and Technology.

Moreso, the researchers were able to conduct an in-depth discussion on the deviation variables based on the supplied priorities and link the findings to the weights and priority levels allocated to these variables based on the results acquired. They check that the outcomes of the models comply with the criteria of meeting the highest priority goals in line with the corresponding weights of the five departments in the Faculty of Science and Technology by discussing these deviational variables.

As a result, they believed that the preemptive goal programming model can be utilized for policy-making in the process of allocating students to academic departments or faculties in the

future. The weighted mean absolute percentage error is then used to compare the outcomes of the preemptive goal programming model to those of the existing allocation. The successful implementation indicates the goal programming model's capacity to meet the academic departments' student intake requirements and goal limitations.

2.3   Summary of Related Literature Reviews

Author & Year	Title & Description	Merit and Demerits
Ojo et al. (2022)	Design and implementation of student project management and allocation system. This project aims to create a web-based system that handles the activities of "student Project Management and Allocation."	An ideal set of programming languages was used in implementing the system. Final-year students have to register on the site themselves which implies that anyone can register on the site and the allocation process is not automated.
Amadi et al. (2021).	Design and Implementation of Students' projects allocation system. This study focuses on building and implementing a software system for allocating student projects (dissertations)	The system allocates a project to the student based on the highest score awarded by the supervisor to the student's proposals and duplication of the thesis was mitigated. The system might lack scalability, high influx of users (students) might make the site slow.

Adamu A., (2020).	Final Year Student Project Allocation Archiving and Management System. This study tackles its stated problems by establishing a final-year student project allocation, archiving, and management system in which the supervisor and student may engage in real-time checkmate for project approval and submission.	Real-time project approval and submission. Security threat as users' information is stored in its session as long as possible unless they are manually logout.
Huda et al. (2020).	E-Assessment in Higher Education: Student's Perspective. This study investigates the use of e-assessment in higher education from the standpoint of students, as well as the students' attitudes toward this approach.	The study is based on the thoughts of a small sample of university students chosen at random, each of whom had a different reaction to e-assessment. Some students acknowledge the need for e-assessment, but they are anxious about technology-based tests since not all of them have the same level of IT proficiency.
Martin and Fanus (2018)	E-Assessment in Higher Education: A Review. The purpose of this research is to give a discussion on e-assessment that focuses on ideas such as tasks that may be obtained through e-	E-assessment can be beneficial provided the assessment is credible and lecturers work together to develop authentic, consistent, transparent, and practicable assessments.

	assessment, benefits and problems of e-assessment, and e-assessment principles.	
Supianto and Khaerudin. (2020).	Web-based project assessment. The goal of this research is to perform a review of the literature on project-based learning and assessment.	The research showed that web-based assessment is more flexible, reduces paper consumption, and data collecting, and encourages e-learning Other hurdles revealed by the research include the visual style of the internet interface, which may pose problems for its users.
Arumugam et al. (2021)	Academic Project Information Management System. This article describes an automated approach for managing final-year projects and also how a web-based automated system would resolve all faults and mistakes while remaining operational offline	As an automated system, it reduces the chances of errors that occur when records are been updated. The implemented system is a local host platform, which is not ideal for any project unless under development.

| Hassan (2016). | A preemptive goal programming for allocating students into academic departments of faculty.

The study focuses on creating a goal programming model to maximize the placement of students in a faculty's academic departments | A major strength of goal programming is its simplicity and ease of use

A debated weakness is the ability of goal programming to produce solutions that are not Pareto efficient |

2.4   Description of the Current System

In the current system which is the manual process of allocating students to supervisors, the project coordinator responsible for the allocation would typically follow a set of steps to ensure that the process is fair and efficient. First, the project coordinator would gather all of the necessary information about the students and the supervisors. For the supervisors, the information might include, their availability and workload. Next, the project coordinator would begin the process of allocating students with supervisors. The project coordinator implements the allocation manually using the Microsoft Excel application package. Once assigning students to supervisors is completed project coordinator would need to communicate the allocation decisions to the students and supervisors, and provide any necessary support or resources to help the students get started with their research projects. Assessment of student records is paper-based where students will make copies of the assessment sheet and present them to the assessor during the seminar and project defense, and the sheets have to be kept so that they can be recorded for all students. Overall, the current system of allocating students to supervisors and assessing student seminar and project defense is the manual method which can be time-consuming and complex, but it can also provide a more personalized and tailored approach to allocating students and supervisors.

2.4.1   Problems Inherent in the Current System

There are several problems inherent in the current system of allocating students to supervisors and assessing student defense the paper-based method:

i.   **Inefficiency**: Manual allocation of students to supervisors and student assessment can be time-consuming and labour-intensive. It may also be prone to errors and omissions.

ii.   **Lack of fairness**: It can be difficult to ensure that the allocation process is fair and unbiased when it is done manually. There may be a risk of favouritism or discrimination, although the electronic method is not totally devoid of this.

2.5   Analysis of the Proposed System

Keeping in mind the aforementioned shortcoming, the suggested approach efficiently addresses the aforementioned issues. The proposed system is an automated web-based system, that automatically generates an unbiased student-to-supervisor allocation record. In the proposed system the students or supervisors don't register on the site as accounts are automatically created with the dataset acquired from the department, the approach was taken to ensure that random people just don't sign up and to reduce the delay of getting the student details. The project coordinator can easily interact with a form and students will be allocated to their respective supervisors; the system is flexible in the sense that allocation details can be modified. The module that takes care of the assessment of the student seminar and project defense ensures that student is allocated to various halls and the specific day of their defense is attached to the allocation details, as well as allocating assessor to their various halls. Each hall has a chief assessor which can only grade students in their specific hall, not forgetting UI/UX; the system took the user interface and experience as one of its most important criteria. With the proposed system student can log in to view their supervisors and supervisors can log in to view the student under their supervision.

2.5.1   Advantages of the New Proposed System

i.   Efficient organization of student information and allocation records.

ii.   Information look-up is easy, fast, and efficient.

iii. Substantially decreases the time needed to allocate students to their respective supervisors, therefore, increasing work efficiency among the project coordinators.

CHAPTER THREE

METHODOLOGY AND DESIGN

3.1   Introduction

A methodology is a process of rigorous study or inquiry, particularly to unearth new facts or information; hence, research methodology should be good enough to enable the achievement of the specified objectives which are achievable using specific components, such as data collection and design procedures and system modeling (use case, activity, and class diagrams). This chapter contains the input/output specifications, and system requirements for the development of a student-to-supervisor allocation and assessment system.

3.2   Methods of Data Collection

Before developing any system, collecting data and facts about the existing system is critical to understand what is going on. This research was carried out using three methods.

i.   Observation of the Work Environment

ii.   Documentation

iii.   Interview

3.2.1   Observation of the Work Environment

This method was employed to acquire information and data for this study by monitoring how the manual system worked. The most evident flaws in the existing system were discovered via detailed inspection. Using the observational approach, the context in which the observation is made can be modified in a variety of ways.

3.2.2   Documentation

Documentation is a type of secondary data collection. This method makes use of journals, manuals, past work, publications, and other sources. This method of data collection is used because it allows for comparison with past studies. This includes the internet, which is a data collection tool. The internet was used to find information on difficult or ambiguous issues.

3.2.3 Interview

The major purpose of using interviews as a data collection approach is to collect data thoroughly and rigorously. The researcher met with the department's project coordinators and got reliable information based on the researcher's queries.

3.3 System Modeling

A system model is a conceptual model which is a result of system modeling that describes and represents a system. It is an interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the development of this modern system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

3.3.1 Use Case Diagrams
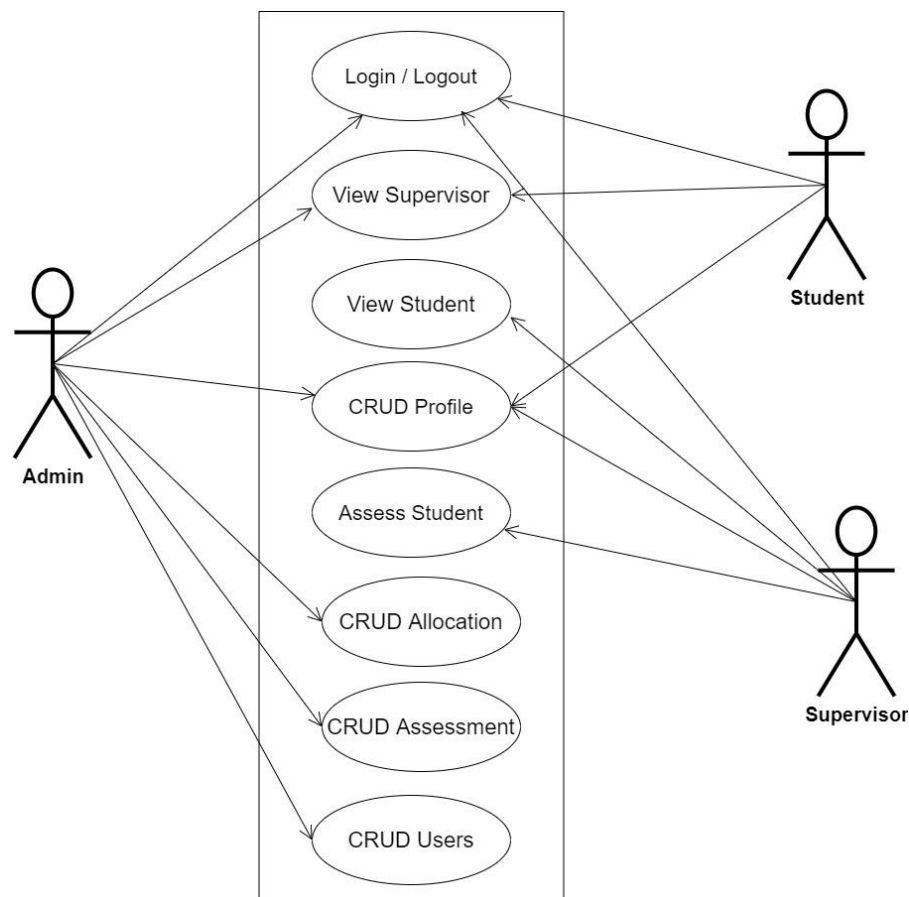
Fig 3.1 System Use Case Diagram

3.3.2 Class Diagrams

Supervisor
- user_id: String
- rank_id: String
- programme_level: String
- title_id: String
- dept_id: String

+ login()
+ viewStudent()
+ assessStudent()
+ updateProfile()
+ logout()

Users
- user_id: String
- username: String
- password: String
- full_name: String
- phone: String
- pics: String
- email: String
- is_super: Bool
- is_superuser: Bool

+ verifyLogin(): bool

Admin
- user_id: String
- rank_id: String

+ login()
+ CRUDStudent()
+ CRUDAssessment()
+ CRUDAllocation()
+ updateProfile()
+ logout()

Assessment
- assess_id: String
- seminar_grd: String
- project_grd: String
- supervisor_grd: String
- dept_id: String
- prog_id: String
- sess_id: String
- stud_id: String
- type_id: String
- assessor_id: String

+ CRUD()

Students
- user_id: String
- programme_id: String
- session_id: String
- dept_id: String

+ login()
+ viewSupervisor()
+ updateProfile()
+ logout()

Allocations
- session_id: String
- prog_id: String
- dept_id: String
- sess_id: String
- stud_id: String
- super_id: String
- type_id: String
- group_id: String

+ CRUD()

Extends

Fig 3.2 System Class Diagram

3.3.3 Activity Diagrams

Login

The process for gaining access to the system is depicted in the diagram below; the email address and password must be accurate to gain access.
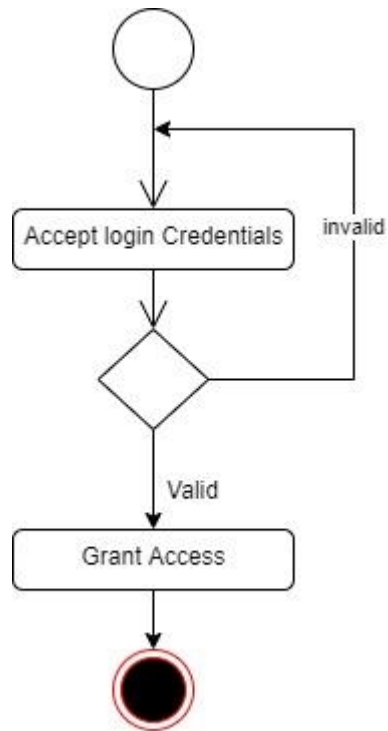
Fig 3.3 System Login Activity Diagram

Create Allocation

The process for automatically allocating students to their supervisors is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the allocation.
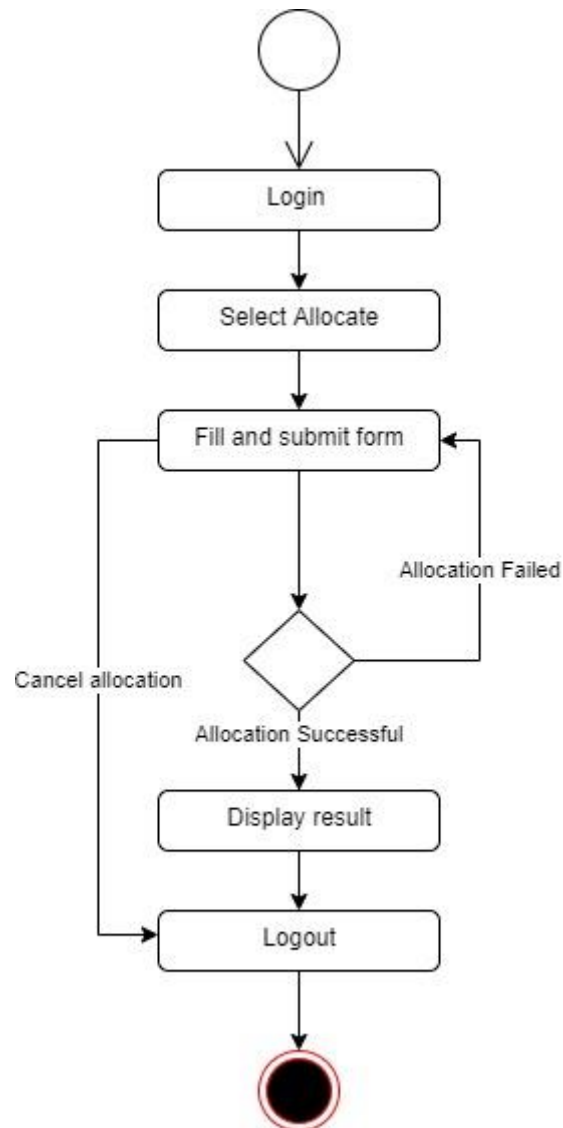
Fig 3.4 Creating Allocation Activity Diagram

Updating / Deleting Allocation

The process for updating/deleting allocation is manually done, as depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the allocation.
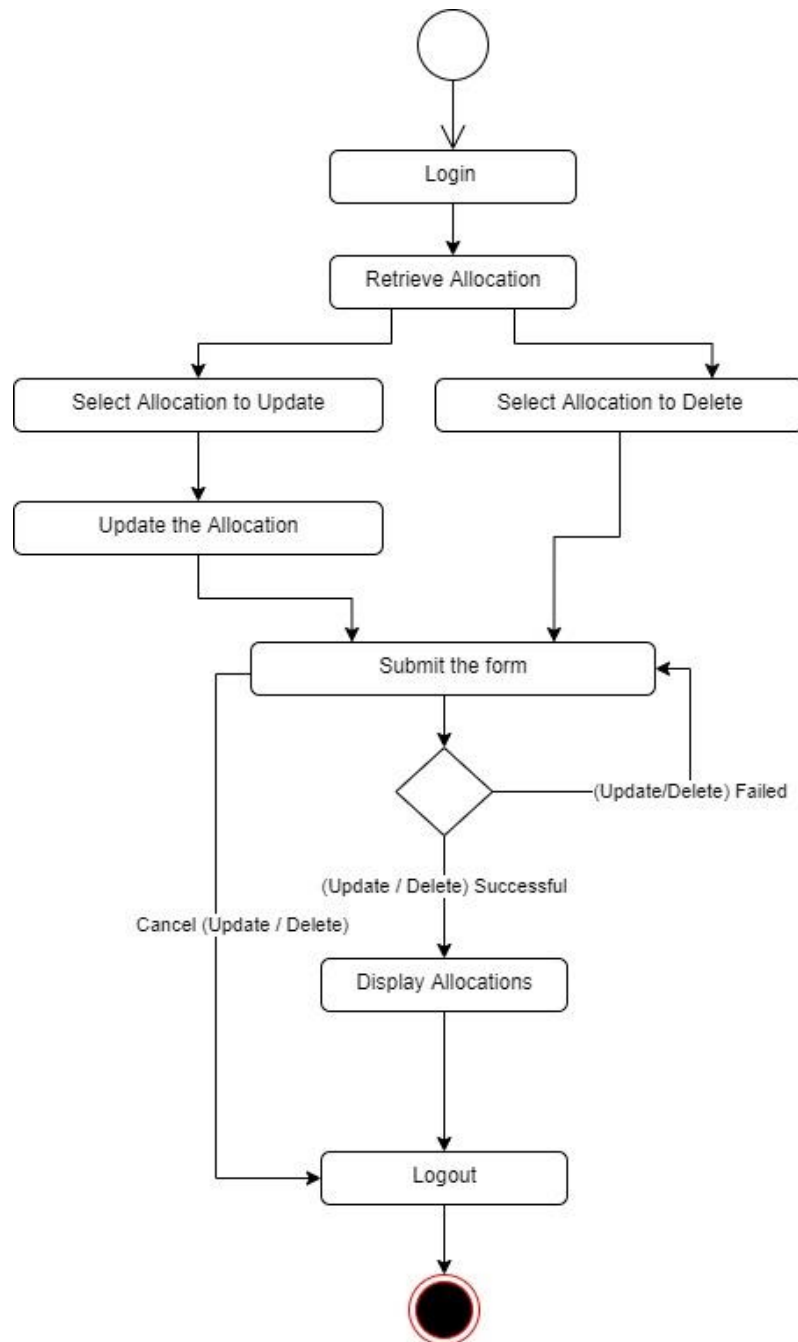
Fig 3.5 Updating / Deleting Allocation Activity Diagram

Creating Assessment

The process for assessing seminar and project defense for students is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the assessment.
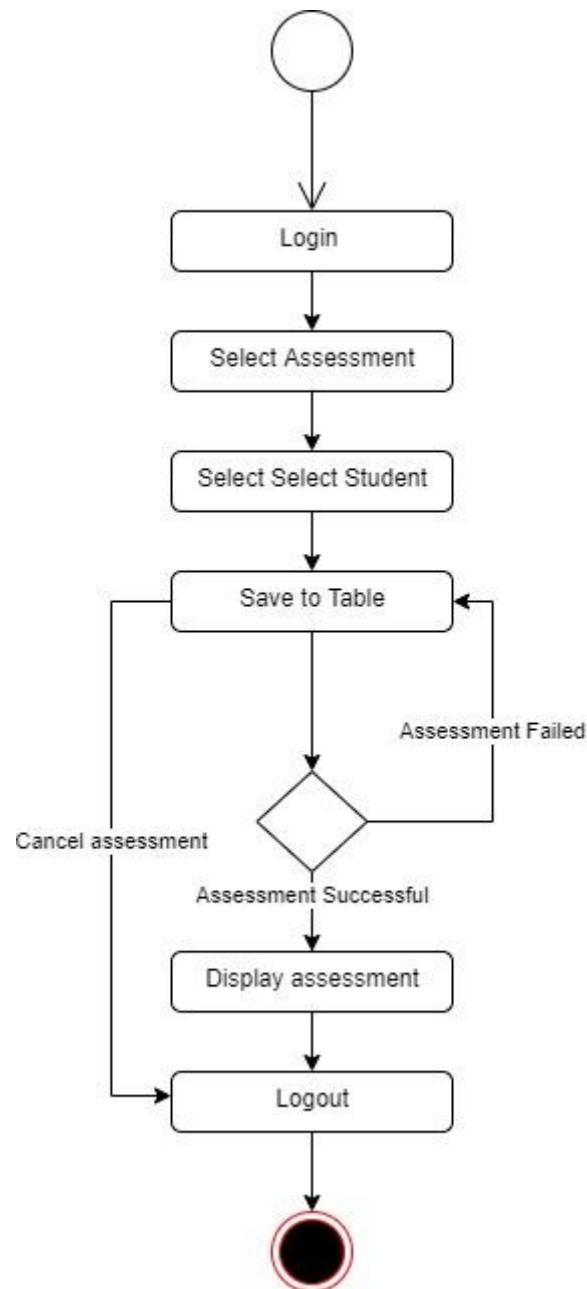
Fig 3.6 Creating Assessment Activity Diagram

Updating / Deleting Assessment

The process for updating / deleting student seminar and project defense assessment is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the assessment.
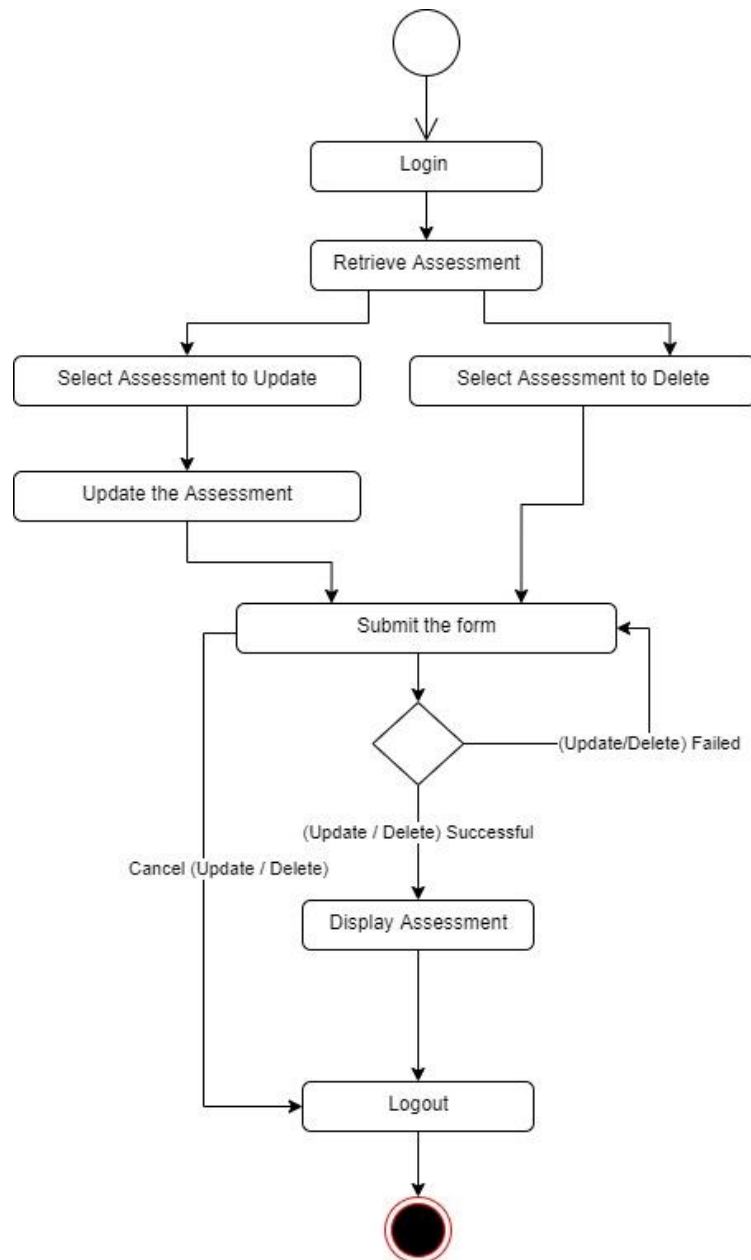
Fig 3.7 Updating / Deleting Assessment Activity Diagram

3.4   Database Design

The following are some of the input specifications used in this project work.

 i.   Users Table: contains the generic information of all system users.

 ii. Allocation Table: contains every system student-to-supervisor allocation information.

 iii. Assessment Table: contains every student assessment information

Table 3.1   Users Input Specification Table

Field Name	Data Type	Null	Key	Length	Description
user_id	Varchar	No	PK	32	Unique string for identifying users
username	Varchar	No		20	Student regNo/Supervisor fileNo
password	Varchar	No		128	User Password
full_name	Varchar	No		60	User full name
phone	Varchar	Yes		11	User phone number
pics	Varchar	Yes		100	User profile picture
email	Varchar	Yes		100	User email address

Table 3.2   Allocation Input Specification Table

Field Name	Data Type	Null	Key	Length	Description
allocation_id	Varchar	No	PK	32	Unique string for identifying allocation
dept_id	BigInt	No		10	Student department reference
prog_id	BigInt	No		10	Student programme (ND/HND) reference
sess_id	Varchar	No		32	student academic session reference
stud_id	Varchar	No		32	Student registration number reference
super_id	Varchar	No		32	Supervisor file number reference
type_id	BigInt	No		10	Student type (Reg/Eve) reference
group_id	BigInt	No		10	Student group number reference

Table 3.3   Assessment Input Specification table

Field Name	Data Type	Null	Key	Length	Description
assess_id	Varchar	No	PK	32	Unique string for identifying assessment
seminar_grd	Int	No		10	Student seminar grade
project_grd	Int	No		10	Student project grade
supervisor_grd	Int	No		10	Student supervisor grade
dept_id	Varchar	No		32	Student department reference
prog_id	BigInt	No		10	Student programme (ND/HND) reference
sess_id	BigInt	No		10	student academic session reference
stud_id	Varchar	No		32	Student registration number reference
type_id	BigInt	No		10	Student type (Reg/Eve) reference
assessor_id	Varchar	No		32	Assessor file number reference

3.5   Output Design

This declares and displays the outcome of the given input. This automated system's output is dependent on its input. The output specification is listed below.

Table 3.4   Users Output Design Table

PROJECT MANAGER

List of the System Registered Users

User_id	Username	Full_name	Phone	Email	Date_registered
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

Table 3.5   Allocation output design table

PROJECT MANAGER

List of Student to Supervisor Allocation

allocation_id	dept_id	prog_id	sess_id	stud_id	super_id	type_id	group_id
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

Table 3.6   Assessment output design table

PROJECT MANAGER

List of Student Defense Assessment

assess_id	dept_id	prog_id	sess_id	stud_id	project_grd	supervisor_grd	seminar_grd
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

3.6   Input & User Interface Design

This shows a visual representation of the system interface; it will be made to be intuitive to use, quick to respond to, and visually appealing. Additionally, it will be properly protected, so authorization will be necessary to view some levels of the contents. A mid-fidelity wireframing application named Draw.io is used to assist with the designs.
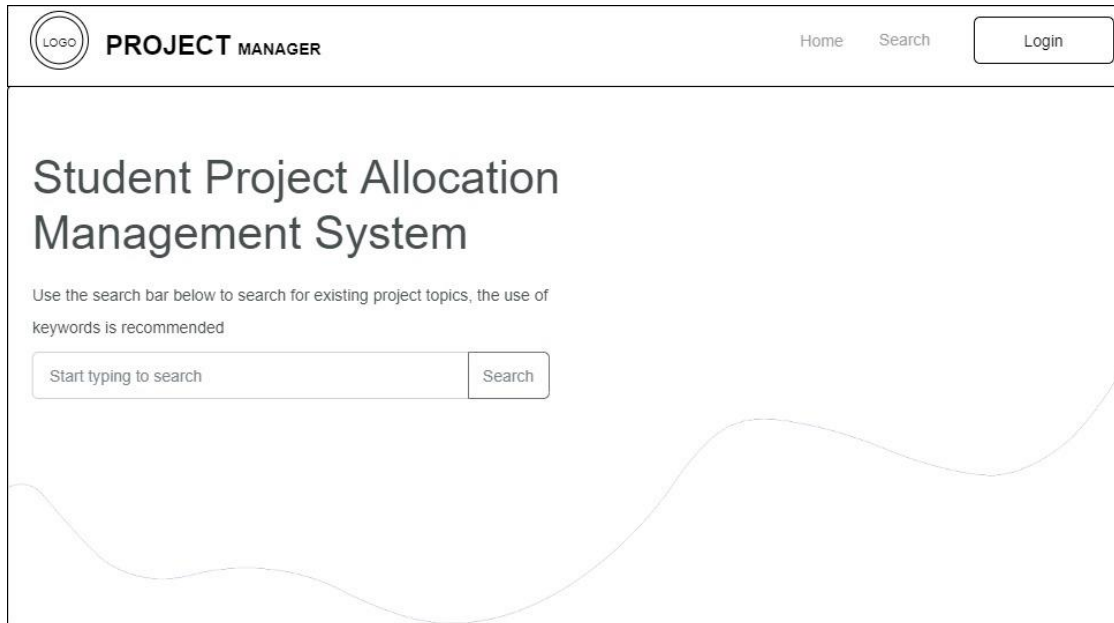
Fig 3.8 Home Page

Fig 3.9 Login Form

28
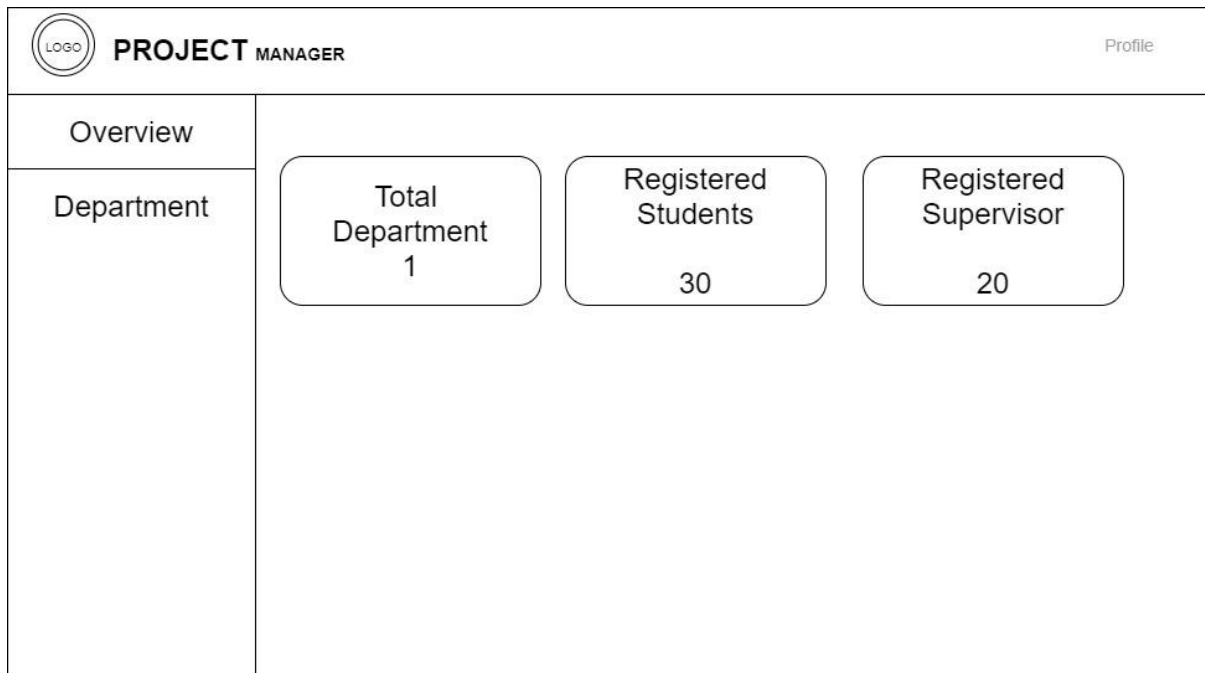
Fig 3.10 System Admin Dashboard
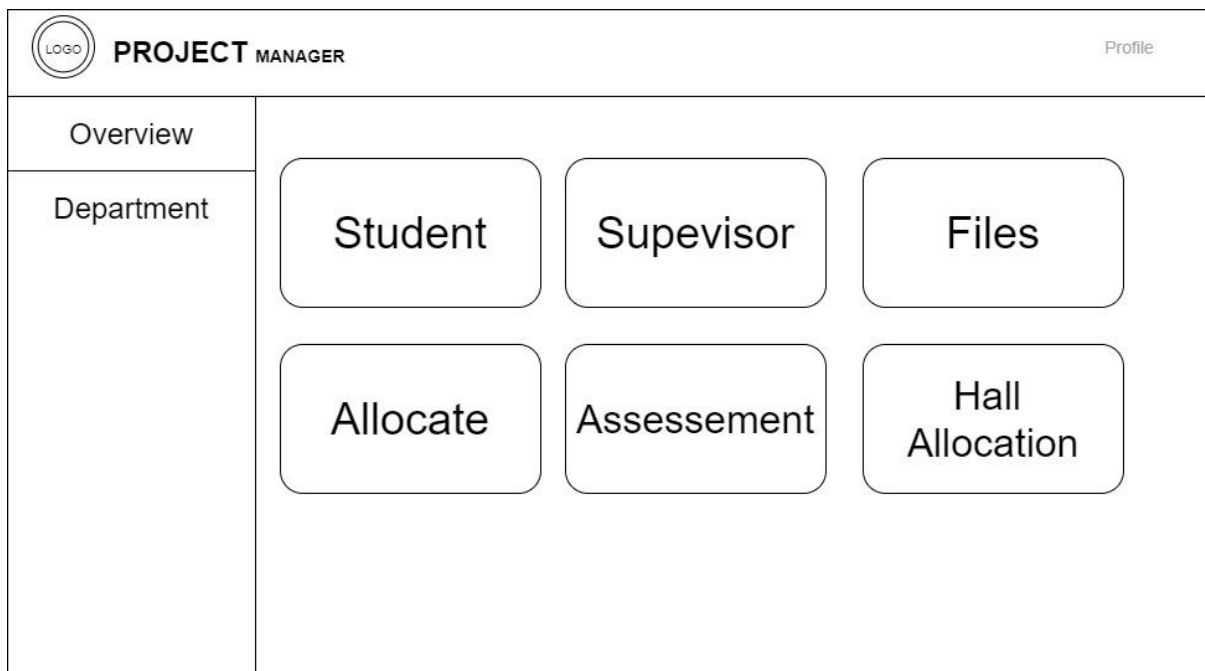
Fig 3.11 Department Dashboard

Fig 3.12 Update Profile Form

Fig 3.13 Defense Assessment Screen

Fig 3.14 Student to Supervisor Allocation Form

3.7   System Requirement

For optimal performance, every software system developed has a specified system requirement on which it has been intended to run. The system requirements, on the other hand, are the minimum hardware and software requirements for the system's smooth functioning.

3.7.1   Hardware Requirement

System Hardware Requirement Include:

    a. Minimum of 2 GB of RAM (Random Access Memory).
    b. Minimum of Intel Dual core processor.
    c. Minimum of 250GB HDD (Hard Disk Drive).

3.7.2 Software Requirement

The software requirements include:

    a. At least Windows 7 OS (Operating System).

    b. Vs. Code IDE installation.

    c. Browsers include Chrome and Firefox.

3.8 Choice of Programming Language

This research work will be a web-based application and will be implemented on a relational database system (SQLite). HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript for the frontend development while Django (Python) will be employed for the backend programming.

CHAPTER FOUR

SYSTEM IMPLEMENTATION EVALUATION

4.1   Introduction

This section provides a concise overview of the implementation process for the new system, ensuring its effective operation. It includes working samples of the newly designed system and outlines the installation procedures.

4.2   System Testing and Evaluation

Testing the developed system is essential for various reasons. It allows us to identify any issues or flaws in the new system and provide solutions to address these problems. In this project, a combination of unit and integration testing was employed to verify the effectiveness and efficiency of the design. The testing process also aimed to ensure that the new system meets its functional requirements and operates without errors.

Unit Testing

This section focuses on conducting unit testing, which involves evaluating individual units or components of the system separately. The purpose is to ensure that each specific phase or component functions correctly and without any issues. Examining these units in isolation makes it possible to verify their proper operation and identify any potential problems or errors.

Integration Testing

Integration testing was employed to test the software, wherein all the components were brought together and operated as a unified system. The objective of this testing phase was to verify the connectivity and proper integration of the various components. By examining the interaction and collaboration between these components, it was ensured that they work seamlessly together and fulfill the intended functionality.

4.3   System Installation

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

a.   Make sure, pip, pipenv, and python3 or greater are installed on the system.
b.   Copy the project folder to any location of your choice.
c.   Open the project folder in Visual Studio Code
d.   On the terminal run "pipenv install -r requirements.txt"
e.   On the terminal run "python manage.py runserver"
f.   Open any browser on the system example Chrome, Microsoft Edge, or Mozilla Firefox.
g.   On the address bar, type http://127.0.0.1 and press the enter key the site should be loaded.

4.4   Security Measures

Since the scope of the application is public, literally all the information is made available to any user, but some functionalities are restricted to the system admin, project coordinators, and supervisors. Functionalities that have to do with creating student accounts, allocation of the student to a supervisor, the assessment of students, etc. are restricted depending on the user type. The restriction is carried out by using passwords.

4.5   Sample Outputs

These descriptions provide a comprehensive overview and visual representation of the program or software. They aim to give a clear understanding of the design by showcasing and illustrating all the interfaces involved.

Homepage

This depicts the homepage which is the introductory page of the website. It is the first page that users typically see when they visit a website's domain or click on its link. The purpose of this page is to provide an overview of the website's content, and features
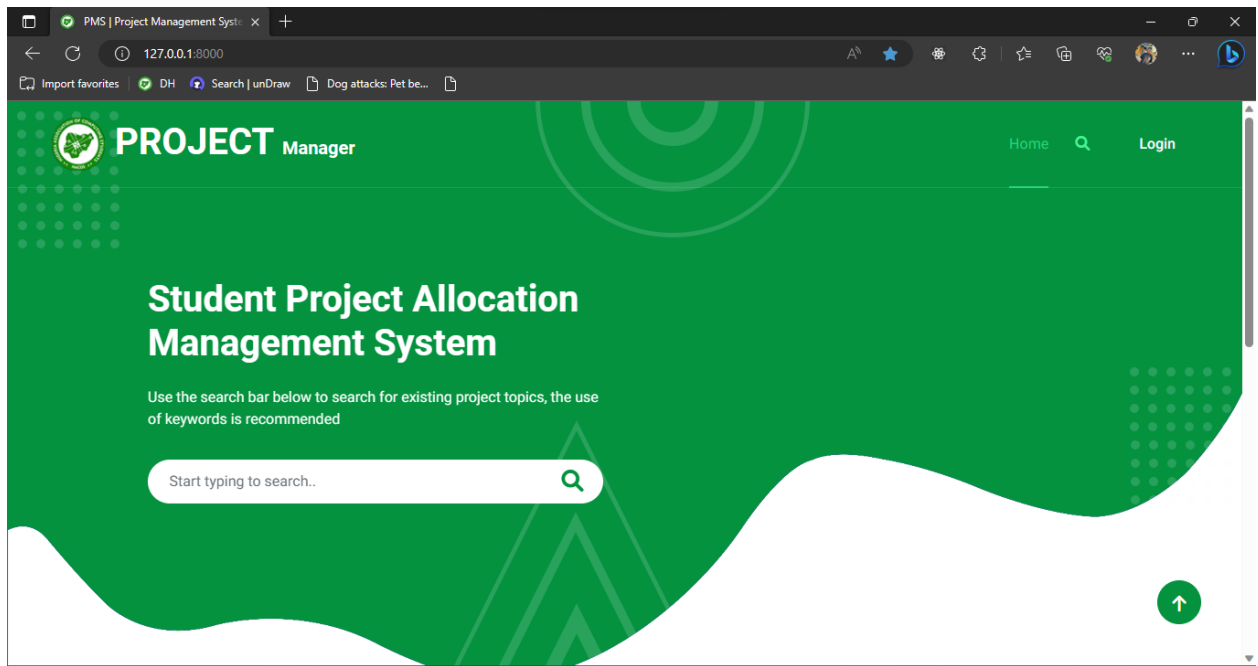
Fig 4.1: Homepage

User Login

This page serves as a gateway that grants access to the system for specific users, including students, system administrators, and supervisors. However, access is only granted if the correct credentials are provided.
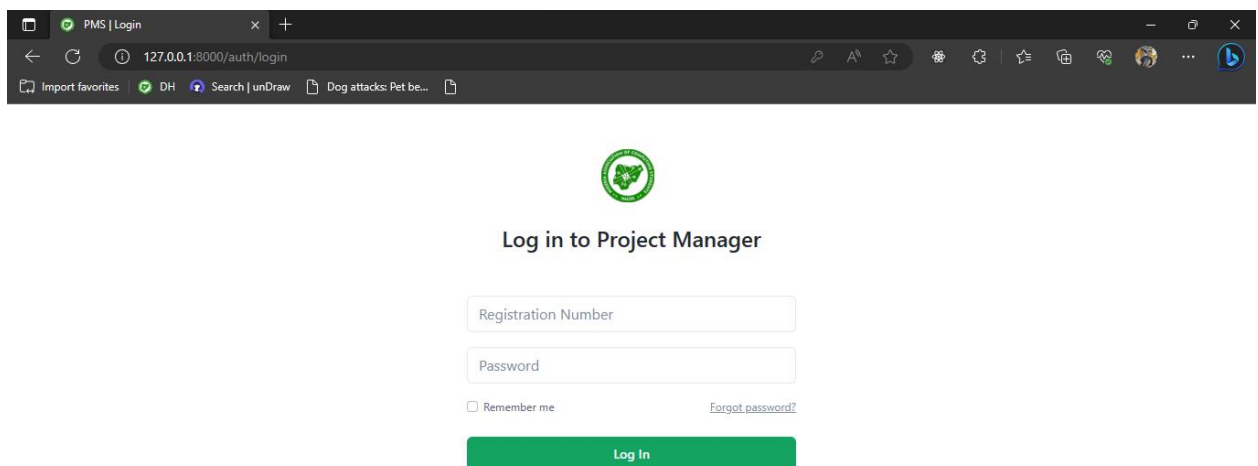
Fig 4.2: User Login

System Admin Dashboard

This is the system admin dashboard, the sidebar shows the available functionality for the user.
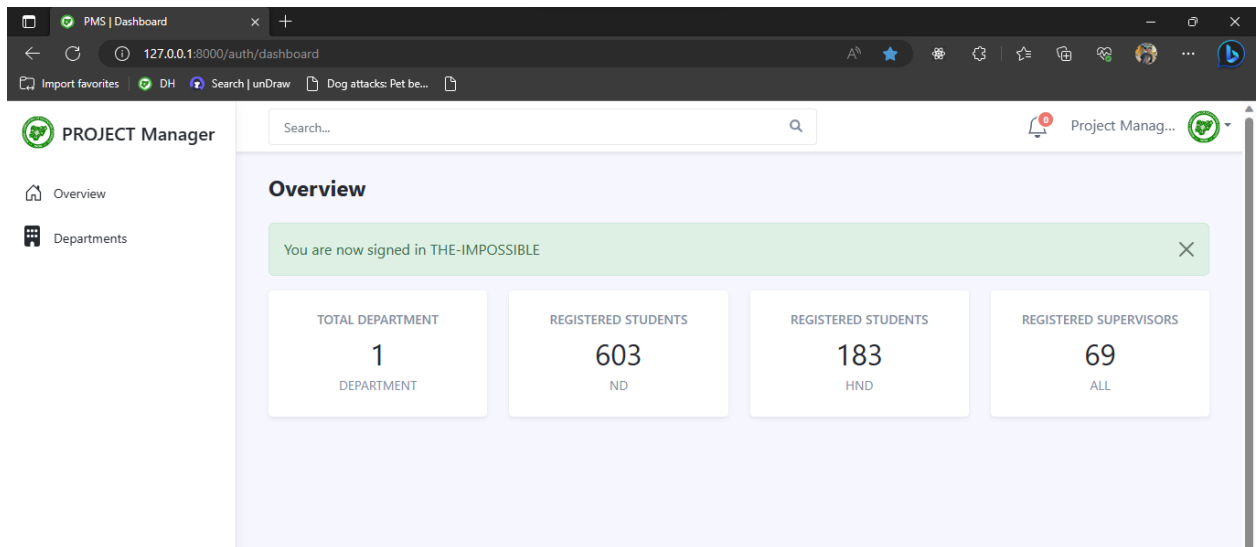
Fig 4.3 System Admin Dashboard

Department Dashboard

The page appears if the department option on the sidebar is clicked upon, showing the different functionalities that can be performed in the department.
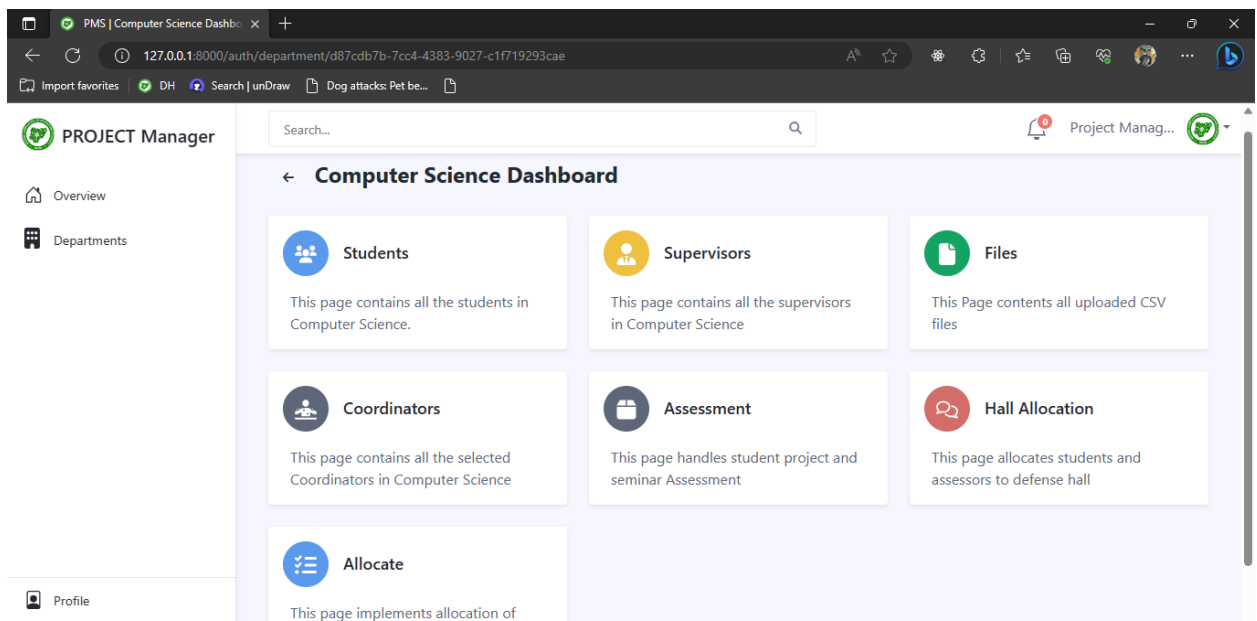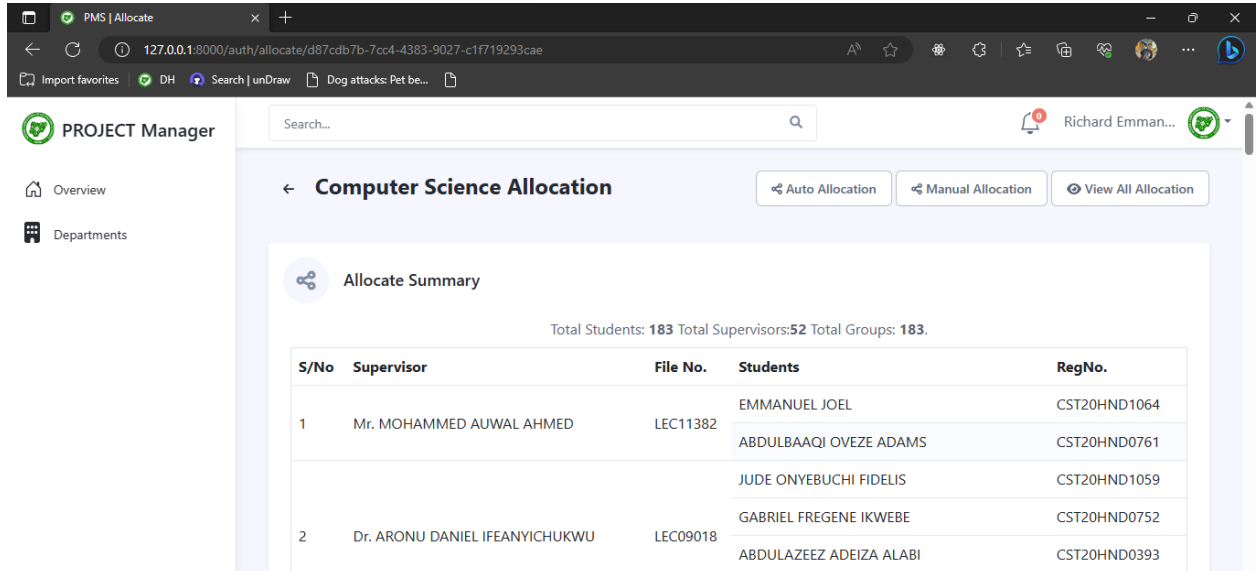
Fig 4.4 Department Dashboard

Allocation Page

This is the page where the admin can allocate students to supervisors either through auto or manual mode and as well view the allocations.

Fig 4.5 Allocation Page

Hall Management Page

This is the page where the admin can create venues, create numbers of defense days, assign students to halls and defense days, and assign assessors to halls

Fig 4.6 Hall Management Page

Student to Venue Allocation

This is the page where the admin can assign students to halls and defense days.
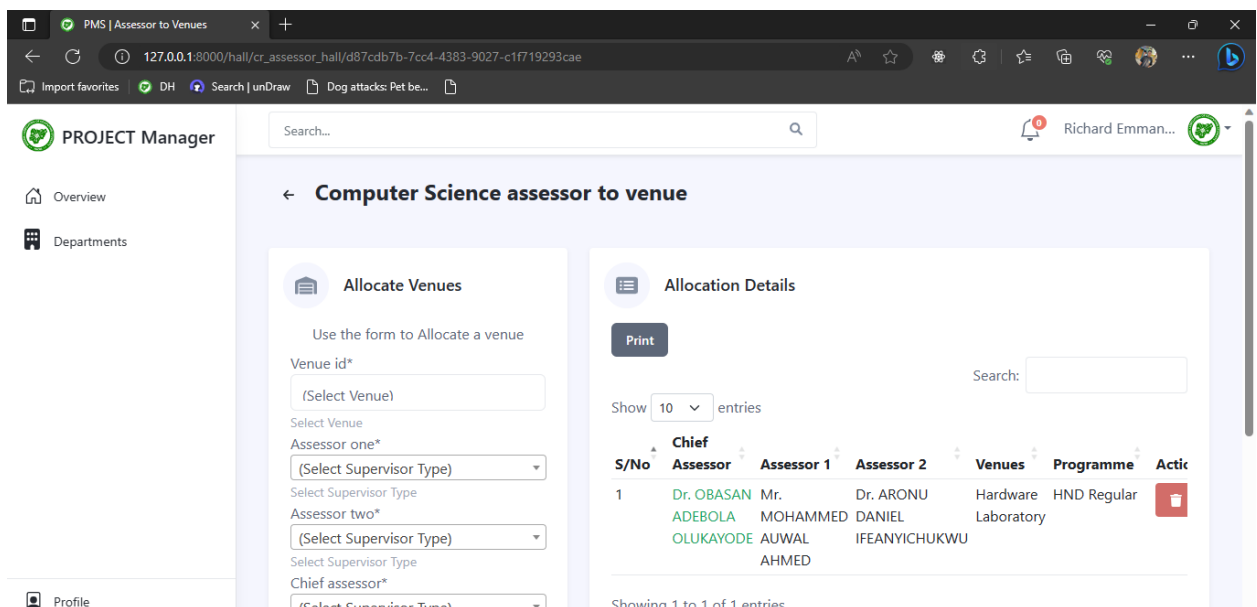
Fig 4.7 Student-To-Venue Allocation

Assessor to Venue Allocation

This is the page where the admin can assign assessors to venues.

Fig 4.8 Assessor to Venue Allocation

Assessor Assessment

This is the page where the assessors or coordinators can grade students either for project or seminar
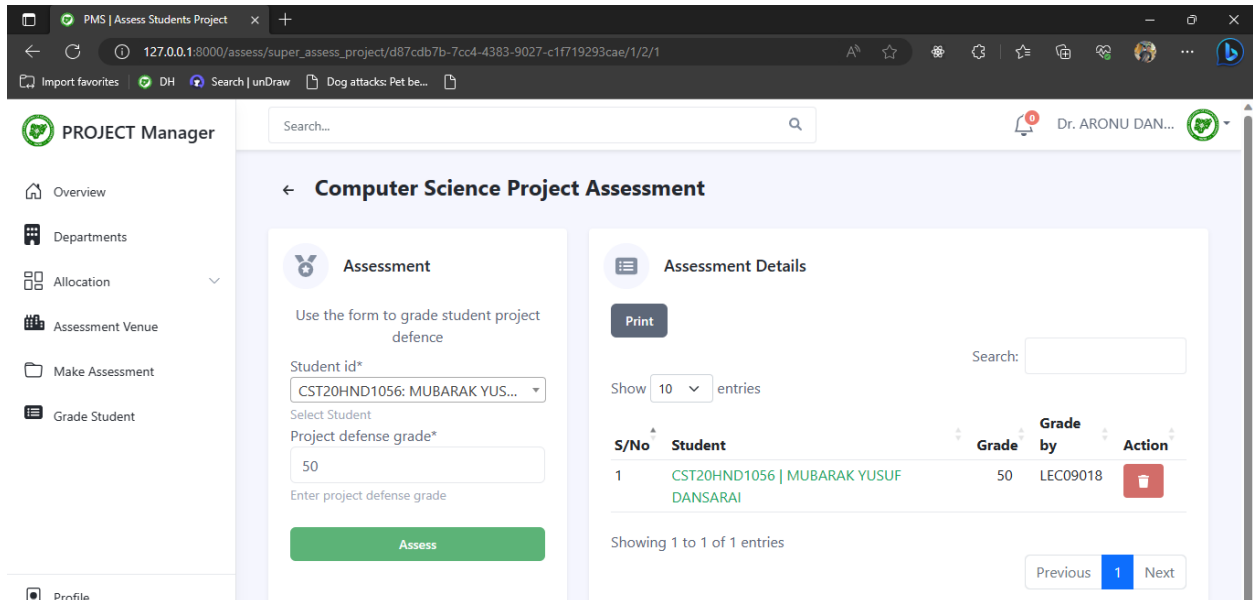

Fig 4.9 Assessor Assessment

Create / Update Coordinators

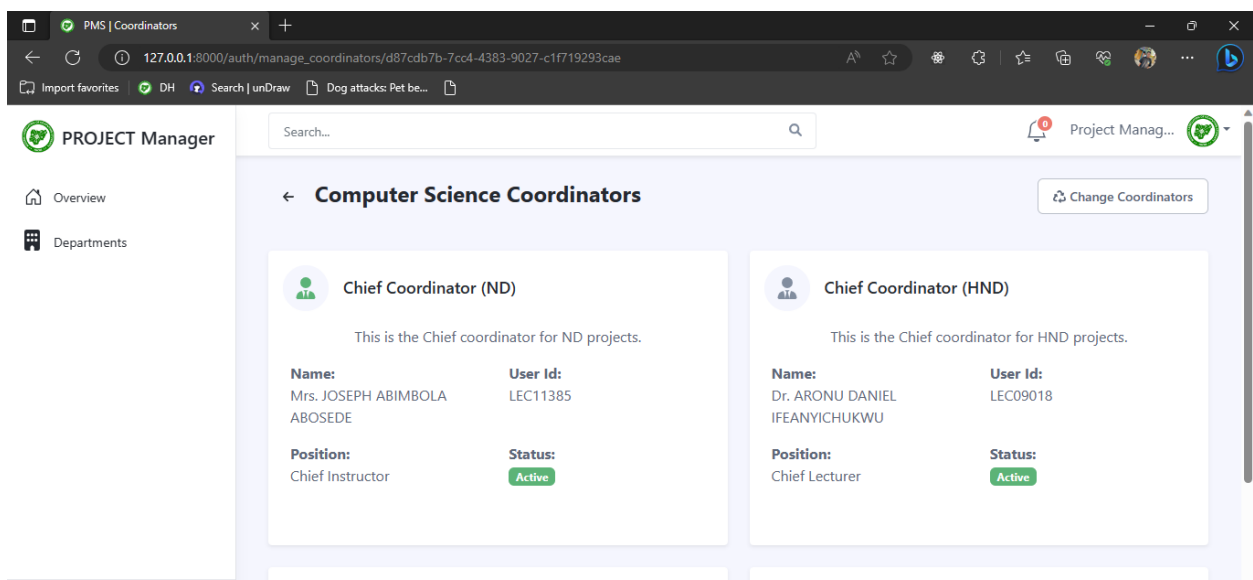This is the page admin can create/update coordinator depending on the programmes

Fig 4.10 Create / Update Coordinators

Student Pages

Student Dashboard

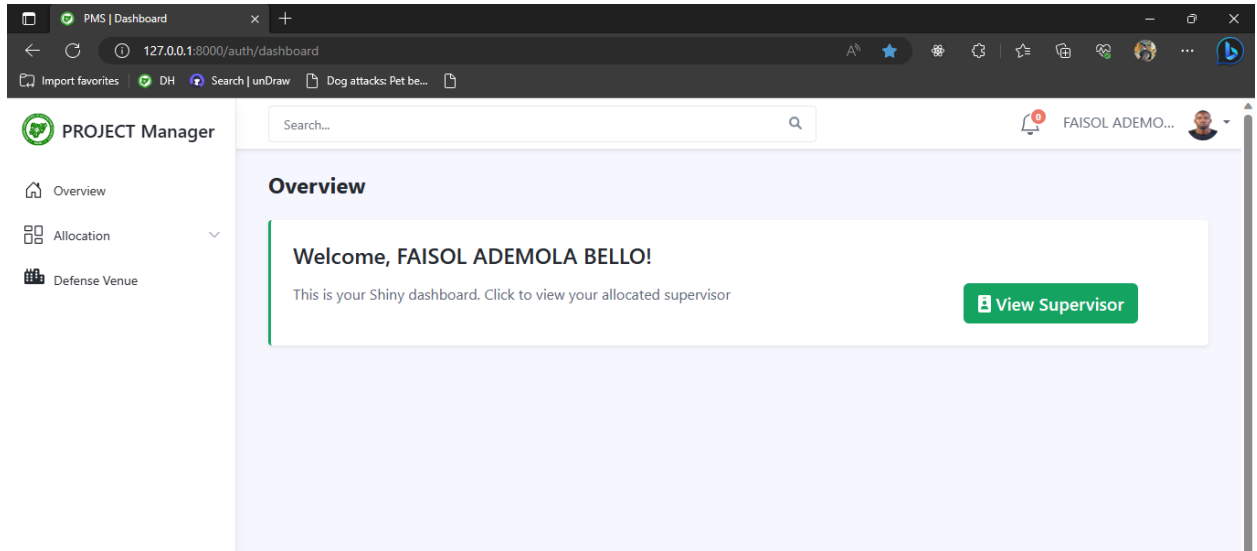This is the student dashboard the sidebar shows the functionalities available for the student.

Fig 4.11 Student Dashboard

View Supervisor

The student can view their allocated supervisors from their dashboard.
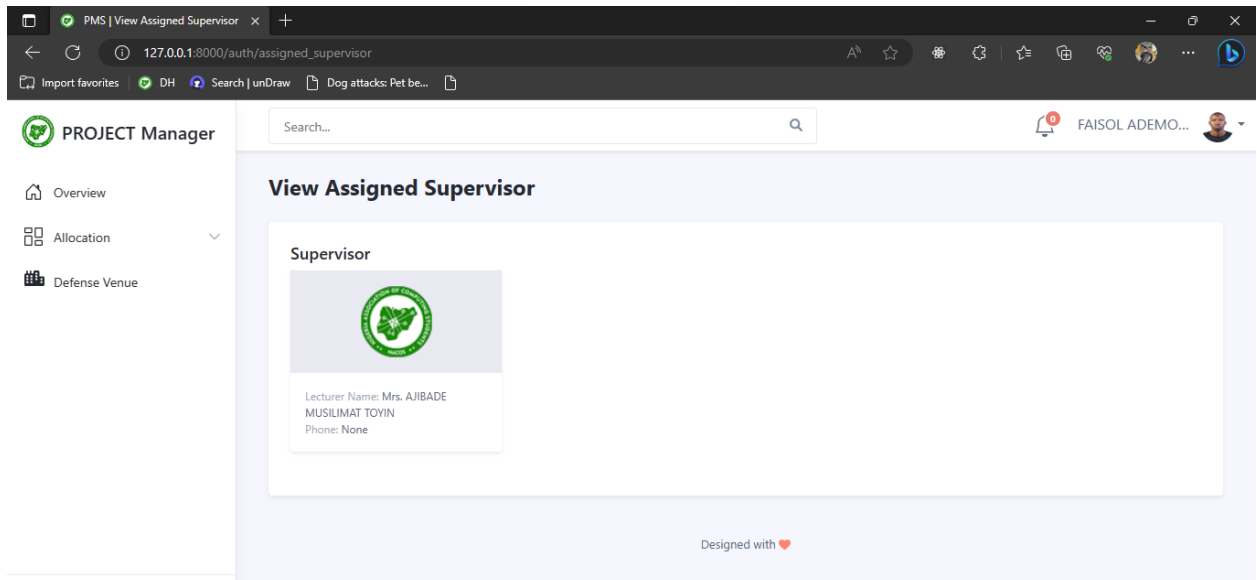

Fig 4.12 View Supervisor

View Venue and Assessors

The student can view their allocated venues along side the venue assessors from their dashboard
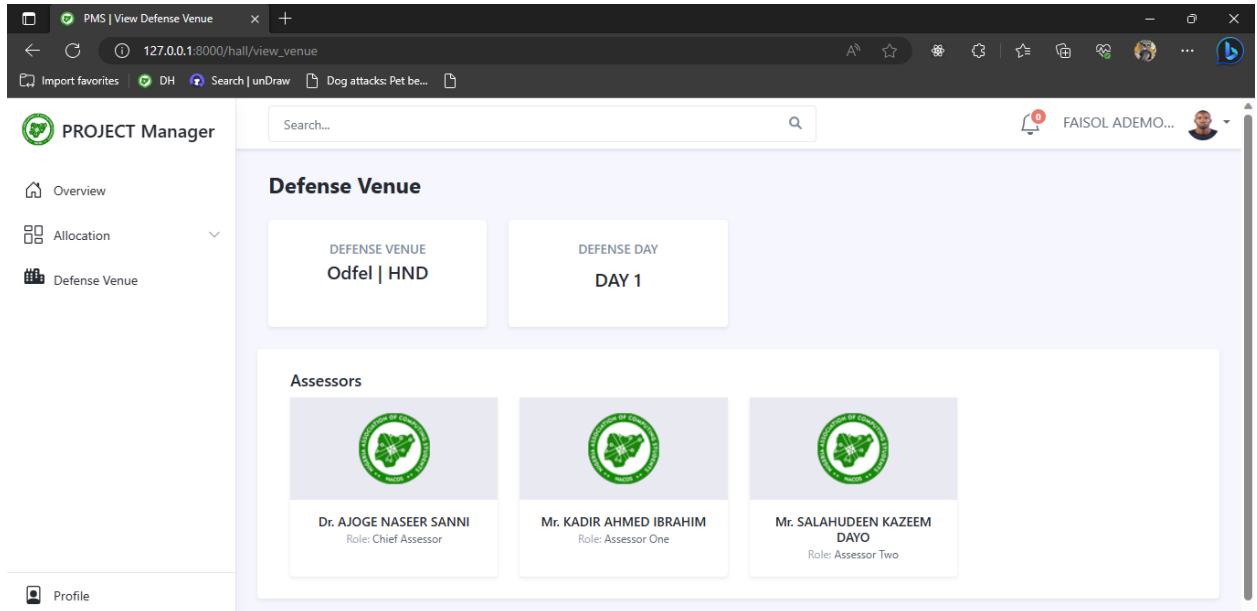
Fig 4.13 View Venue and Assessors

Profile Page

The logged in user can update their profile or decide to change their password
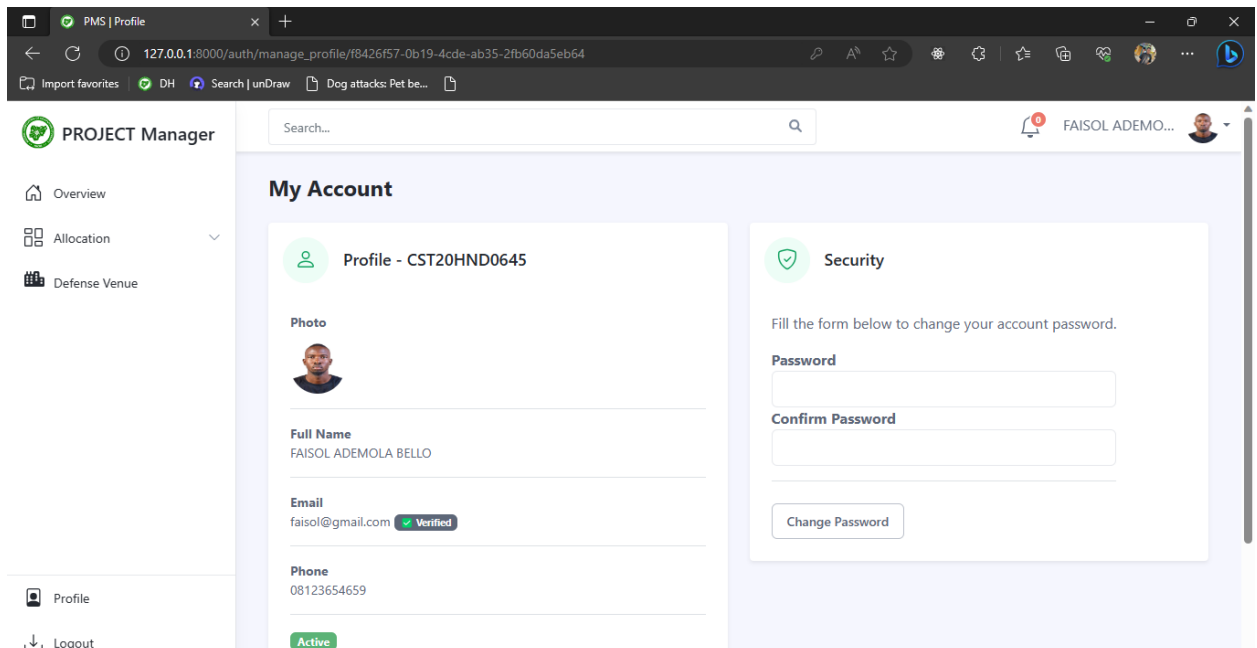
Fig 4.14 Profile Page

Supervisors Page

Supervisor Dashboard

This is the supervisor dashboard the sidebar shows the functionalities available for the supervisor

Fig 4.15 Supervisor Dashboard

View Assigned Student

The supervisor can view the students that are assigned to them

Fig 4.16 Supervisor Dashboard

Grade Project Student

The supervisor can grade their project students

Fig 4.17 Grade Project Student

View Assessment Venue

The supervisors can view their assessment venues along side other assessors

Fig 4.18 View Assessment Venue

CHAPTER FIVE

SUMMARY CONCLUSION AND RECOMMENDATION

5.1   Summary

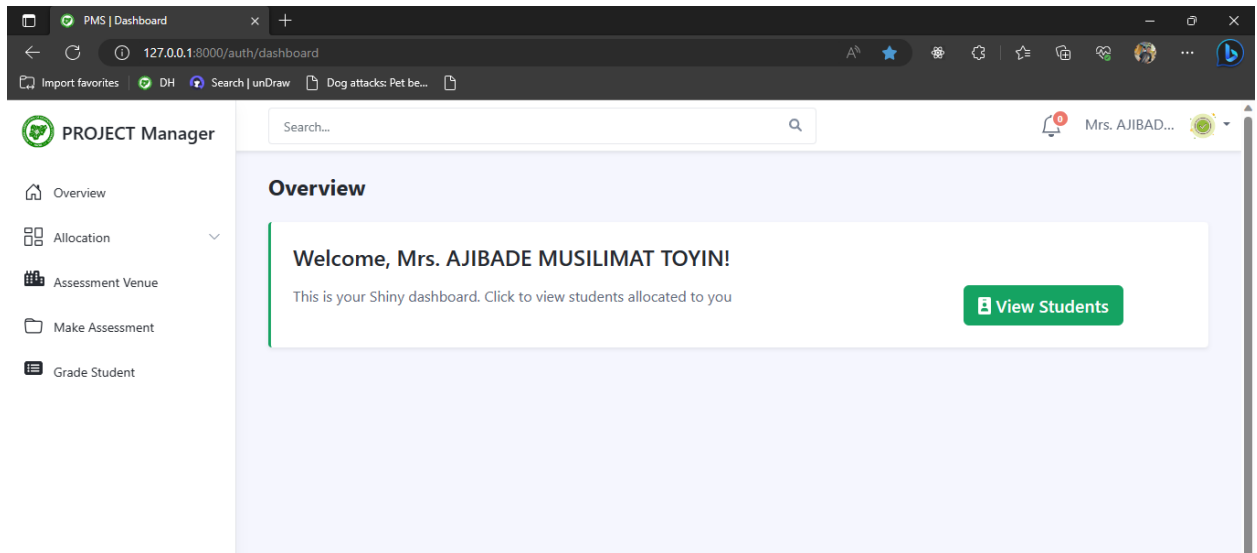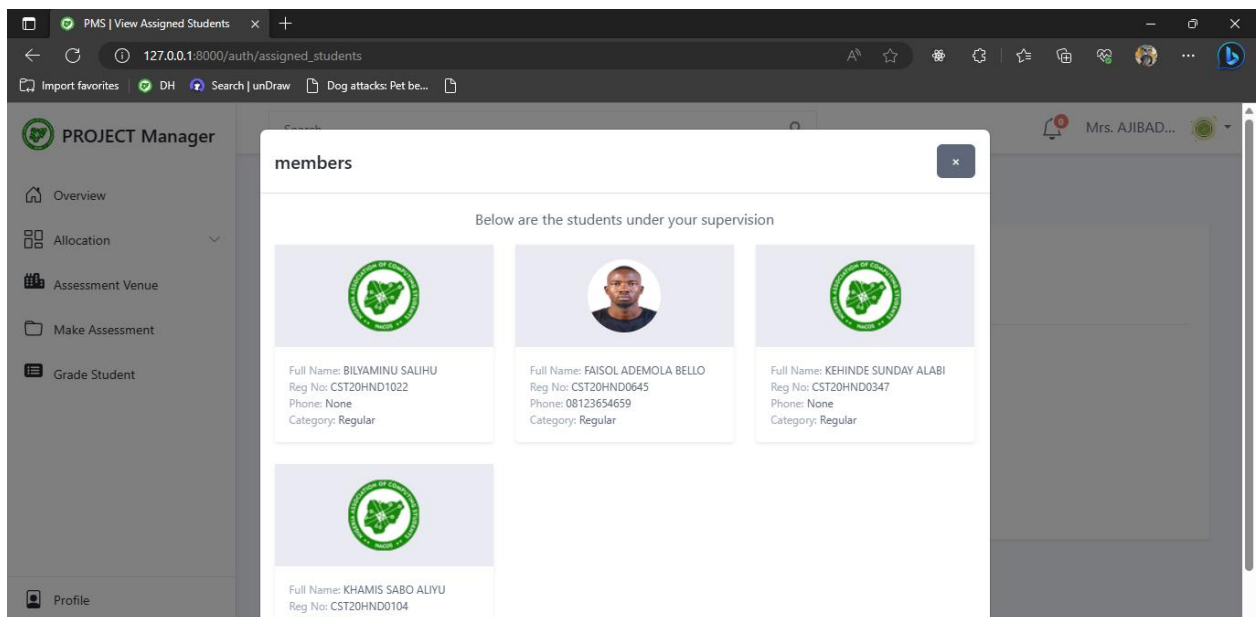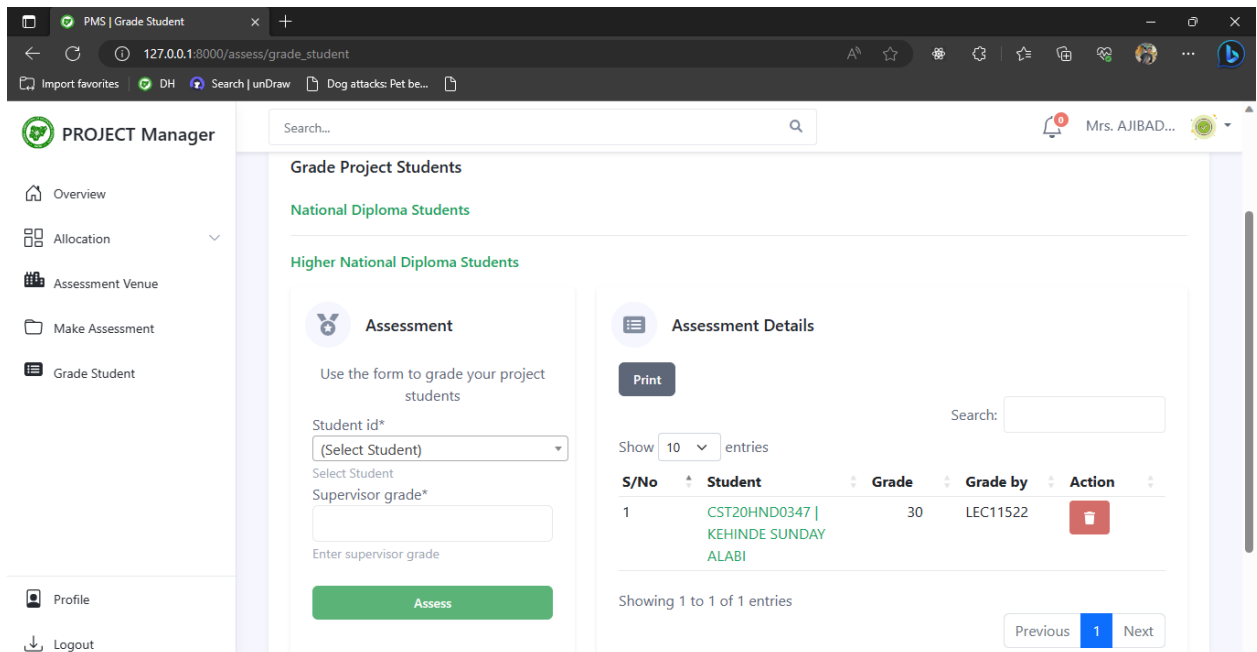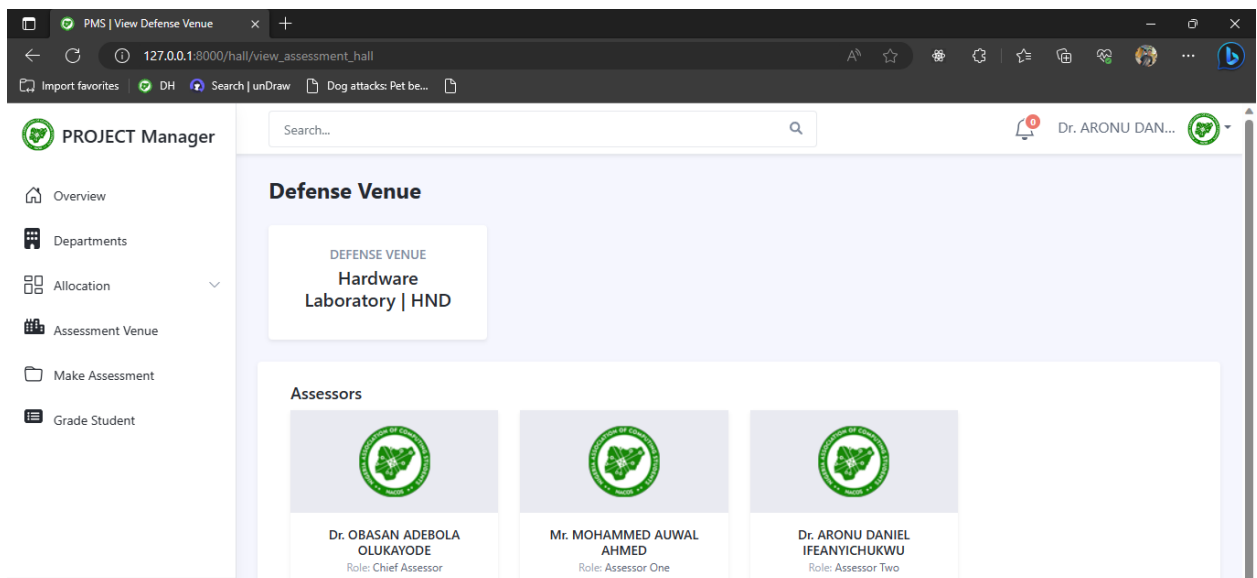This project focuses on the development of a student-to-supervisor allocation and assessment system for the Computer Science department at Kaduna Polytechnic. The system aims to streamline the process of assigning students to supervisors and conducting assessments for student projects and seminar defense. By automating the allocation process and providing a user-friendly web interface, the system enhances work efficiency, improves communication between students and supervisors, and simplifies the assessment procedure. The research utilizes modern technologies and frameworks to create a secure and efficient platform for managing student-supervisor relationships, facilitating effective project management and data handling. This project is significant as it reduces time wastage, improves documentation, and enhances overall productivity in the department's project assessment procedures.

5.2   Conclusion

In conclusion, this project has successfully developed a student-to-supervisor allocation and assessment system for the Computer Science department at Kaduna Polytechnic. The system automates the allocation process and provides a user-friendly web interface, improving efficiency and communication between students and supervisors. By streamlining the assessment procedure, the system reduces paperwork and enhances documentation, resulting in more accurate grading. It significantly reduces time wastage in assigning students to supervisors and facilitates effective project management. Overall, this project has made a valuable contribution to improving the project assessment procedures at Kaduna Polytechnic, increasing productivity, and enhancing the overall experience for students, supervisors, and assessors.

5.2   Recommendation

Based on the project findings and outcomes, the following recommendations are suggested for further improvement and enhancement:

i.   Continuous System Monitoring: It is recommended to establish a process for ongoing system monitoring and maintenance. Regularly monitoring the system's performance, addressing any issues or bugs, and implementing necessary updates will ensure its optimal functionality and reliability.

ii. Implementation and Deployment: Proceed with the implementation and deployment of the student-to-supervisor allocation and assessment system to the Computer Science department at Kaduna Polytechnic. Ensure thorough testing and validation before making it operational.

iii. Scalability and Flexibility: Consider the future growth and expansion of the system by designing it to be scalable and flexible. This will allow for easy integration of additional features, accommodate a larger number of users, and adapt to changing requirements.

By considering these recommendations, the student-to-supervisor allocation and assessment system can be continuously improved, ensuring its long-term effectiveness and alignment with the evolving needs of the Computer Science department at Kaduna Polytechnic.

REFERENCES

Arumugam, V., Singh, P., Padhiyar, K., Manek, R., & Sayyad, S. (2021). Academic Project Information Management System. *Institute of Electrical and Electronics Engineers Inc*. https://doi.org/10.1109/ASIANCON51346.2021.9544565

Amadi, A. N., Ololo, E. C. (2021); Design and Implementation of Students' projects allocation System. *International Journal of Scientific and Research Publications (IJSRP) 11*(6) (ISSN: 2250-3153), https://dx.doi.org/10.29322/IJSRP.11.06.2021.p11424

Django. (2022). "*The web framework for perfectionists with deadlines*" https://www.djangoproject.com/

Hassan, N. (2016). A preemptive goal programming for allocating students into academic departments of a faculty. *International Journal of Mathematical Models and Methods in Applied Sciences*, 10, 166–170.

Huda, S. S. M., Kabir, Md. & Tanvir S. (2020). E-Assessment in Higher Education: Student's Perspective. *International Journal of Education and Development using Information and Communication Technology 16*(2), pp. 250-258

Martin, A., Fanus, V. (2018) E-Assessment in Higher Education: *A Review. International Journal of Business Management and Economic Research (IJBMER), 9*(6),2018, 1454-1460

Ojo, T., M., Shuaibu, N,. Nasir B., S., (2022). Design and Implementation of Student Project Management and Allocation system: Kaduna Polytechnic, Department of Computer Science

Supianto, K., & Khaerudin. (2020). Web-based project assessment. *International Journal of Psychosocial Rehabilitation, 24*(8). https://doi.org/10.37200/IJPR/V24I8/PR280689

APPENDIX

```python
# My Django imports
from django.shortcuts import render, redirect, reverse
from django.views import View
from django.core.files.storage import default_storage
import csv, io, codecs, random, os
from pprint import pprint
from django.views.generic import ListView
from django.contrib.auth import authenticate, login, logout, get_user
from django.contrib import messages
from django.core.exceptions import ObjectDoesNotExist
from django.contrib.auth.mixins import LoginRequiredMixin
from django.contrib.messages.views import SuccessMessageMixin
from django.views.generic import ListView, DetailView, CreateView, UpdateView,
DeleteView
from django.http import HttpResponse
from django.core.exceptions import ValidationError
from django.conf import settings
from django.contrib.staticfiles.storage import staticfiles_storage
from django.contrib.auth.hashers import make_password
from django.db import IntegrityError
from django.utils.decorators import method_decorator
from django.core import serializers
import pickle
#Email
from django.utils.http import urlsafe_base64_decode, urlsafe_base64_encode
from django.contrib.sites.shortcuts import get_current_site
from PAS_auth.utils import EmailThread, email_activation_token, Email
from django.utils.encoding import force_bytes, force_str, DjangoUnicodeDecodeError

from PAS_auth.form import (
    UserForm,
    UpdateProfileForm,
)
from PAS_assessment.models import (
    Assessment,
)

from PAS_auth.decorator import *

PASSWORD = '12345678'
SPLIT = 6
from PAS_hallAllocation.views import render_to_pdf

class EnforceAuth(LoginRequiredMixin):
    login_url = 'auth:login'
```

```python
# Create your views here.
class LoginView(View):
    def get(self, request):
        return render(request, 'auth/login.html')

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

        if username and password:
            # Authenticate user
            user = authenticate(request, username=username.upper(), password=password)
            if user:
                if user.is_active:
                    login(request, user)
                    messages.success(request, f'You are now signed in {user}')
                    nxt  = request.GET.get('next', None)
                    if user.is_super:
                        try:
                            request.session['dept_id'] =
str(SupervisorProfile.objects.get(user_id=user).dept_id.dept_id)
                        except ObjectDoesNotExist:
                            request.session['dept_id'] = None

                    if nxt is None:
                        return redirect('auth:dashboard')
                    return redirect(self.request.GET.get('next', None))
                else:
                    messages.warning(request, 'Account not active contact the administrator')
                    return redirect('auth:login')
            messages.warning(request, 'Invalid login credentials')
            return redirect('auth:login')
        else:
            messages.error(request, 'All fields are required!!')
            return redirect('auth:login')
class LogoutView(LoginRequiredMixin, View):
    login_url = 'auth:login'
    @method_decorator(only_authenticated_users)
    def post(self, request):
        logout(request)
        messages.success(request, 'You are successfully logged out, to continue login again')
        return redirect('auth:login')
class ResetPasswordView(View):
    def get(self, request):
        return render(request, 'auth/password_reset.html')

    def post(self, request):
        email = request.POST.get('email').lower()
        if email:
            user = User.objects.filter(email=email)
```

```python
        if user.exists():
            current_site = get_current_site(request).domain
            data = user[0]
            user_details = {
                'fullname':data.get_fullname(),
                'email': data.email,
                'domain':current_site,
                'uid': urlsafe_base64_encode(force_bytes(data.user_id)),
                'token': email_activation_token.make_token(data),
            }
            Email.send(user_details, 'reset')
            messages.success(request, 'A mail has been sent to your mailbox to enable you
reset your password!')
        else:
            messages.error(request, "Email address doesn't exist!")
    return render(request, 'auth/password_reset.html')
class ResetPasswordActivationView(View):
    def get(self, request, uidb64, token):
        context = {
            'uidb64':uidb64,
            'token':token
        }
        user_id = force_str(force_bytes(urlsafe_base64_decode(uidb64)))
        try:
            user = User.objects.get(user_id=user_id)
            if email_activation_token.check_token(user, token):
                messages.info(request, 'Create a password for your account!')
                return render(request, 'auth/complete_password_reset.html', context)
            else:
                messages.info(request, 'Link broken or Invalid reset link, Please Request a new
one!')
                return redirect('auth:reset_password')


        except User.DoesNotExist:
            messages.error(request, 'Oops User not found, hence password cannot be changed,
kindly request for a new link!')
            return redirect('auth:reset_password')

    def post(self, request, uidb64, token):
        user_id = force_str(force_bytes(urlsafe_base64_decode(uidb64)).decode())
        context = {
            'uidb64':uidb64,
            'token':token
        }
        try:
            user = User.objects.get(user_id=user_id)
            password1 = request.POST['password1']
            password2 = request.POST['password2']

            if(password1 != password2):
```

```python
            messages.error(request, 'Password don\'t match!')
            return render(request, 'auth/complete_password_reset.html', context)

        if(len(password1) < 6):
            messages.error(request, 'Password too short!')
            return render(request, 'auth/complete_password_reset.html', context)

        user.set_password(password1)
        user.save()
        messages.success(request, 'Password Changed you can now login with new
password')

        return redirect('auth:login')


    except User.DoesNotExist:
        messages.error(request, 'Oops user does not exist!')
        return redirect('auth:reset_password')
class DashboardView(LoginRequiredMixin, View):
    login_url = 'auth:login'
    @method_decorator(has_updated)
    def get(self, request):

        context = {}

        if request.user.is_staff:
            prog_nd = Programme.objects.get(programme_title="ND")
            prog_hnd = Programme.objects.get(programme_title="HND")
            dept = Department.objects.all().count()
            nd = StudentProfile.objects.filter(programme_id=prog_nd).count()
            hnd = StudentProfile.objects.filter(programme_id=prog_hnd).count()
            sup = SupervisorProfile.objects.filter().count()

            context = {
                'dept': dept,
                'nd': nd,
                'hnd': hnd,
                'sup': sup,
            }

        return render(request, 'auth/dashboard.html', context)
```