Institute of Information Technology
Quaid-i-Azam University Islamabad.

Outcome Based Education (Web Application)

By

Muhammad Afraz 09162013004

Fazal Abbas 09162013013

Supervisor

Supervisor Name: Dr Adeel Anjum

Bachelor of Science in Information Technology (2020-2023)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

Institute of Information Technology
Quaid-i-Azam University Islamabad.

Outcome Based Education (Web Application)

A project presented to
Institute of Information Technology
Quaid-i-Azam University, Islamabad

In partial fulfillment
of the requirement for the degree of

Bachelor of Science in Information Technology (2020-2023)

By

Muhammad Afraz **09162013004**
Fazal Abbas **09162013013**

DECLARATION

We hereby state that, the software nor any part of it has been copied from any source. The software and accompanying report have been solely developed through our personal efforts. If any part of the project is found to be plagiarized, we accept full responsibility for the consequences. None of the work presented has been previously submitted for any other academic degree or qualification at any university or educational institution.

Muhammad Afraz

Fazal Abbas

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (IT) "Outcome Base Education" was developed by **Muhammad Afraz (09162013004)** and **Fazal Abbas (09162013013)** under the supervision of "Dr Adeel Anjum" and that in (their/his/her) opinion; it is fully adequate, in scope and quality for the degree of Bachelor of Information Technology.

Supervisor

External Examiner

Director of the Department
(Department of Information Technology)

Executive Summary

The Outcome Based Education project represents a major milestone in the field of education technology. This web-based application is designed to help students, educators, and institutions reach their full potential by providing a platform that allows for personalized, outcome-oriented learning. By shifting the focus from traditional teaching methods to a more student-centered approach, the Outcome Based Education project will help students develop the critical thinking and problem-solving skills they need to succeed in the modern workforce.

The application will be user-friendly and accessible, with a range of interactive tools and resources to support students as they work towards their learning goals. With real-time tracking and reporting capabilities, educators will be able to monitor student progress and provide personalized feedback and support. This will allow for a more collaborative learning environment, where students and educators can work together to achieve success.

In addition to its cutting-edge technology, the Outcome Based Education project will also be highly scalable, making it ideal for educational institutions of all sizes. Whether you're a small start-up or a large university, the Outcome Based Education project has the potential to transform the way you approach education. By providing students with a more engaging and effective learning experience, and educators with a powerful tool to measure progress, the Outcome Based Education project is set to become a leading platform in the education technology space.

Acknowledgement

We are incredibly thankful to Almighty Allah for bestowing upon us a small measure of His boundless wisdom and knowledge. This allowed us to undertake and successfully complete this challenging project. Without this gift, our task would have been much more difficult.

We would like to express our sincere gratitude to our project supervisor for their constant guidance, supervision, and advice throughout this project. Without their unwavering support, the successful completion of this project would have been uncertain. We are grateful for their encouragement and the invaluable contributions they made to our project.

Moreover, we would like to extend our gratitude to our parents and family, who have always been a source of encouragement and support for us. They have instilled in us the values of honesty and hard work, which were instrumental in helping us complete this project. Their love and support have been truly invaluable to us.

Muhammad Afraz

Fazal Abbas

Abbreviations

SRS	Software Require Specification
CO	Course Outcome
PO	Program Outcome
PEO	Program Educational Outcome

Table of Contents

1 Introduction

Outcome-Based Education (OBE) is a modern approach to teaching and learning that focuses on achieving specific learning outcomes. In this approach, the curriculum is designed to help students acquire specific knowledge, skills, and attitudes that are relevant to their future careers. To support this approach, an Outcome-Based Education Web-based Application is a valuable tool that can assist educators in designing and delivering an effective curriculum.

This web-based application is designed to provide a comprehensive and user-friendly platform that allows educators to create and manage lesson plans, assessments, and other educational resources. These resources are aligned with the desired learning outcomes, providing students with a clear understanding of what they need to achieve. Additionally, the application provides students with a personalized learning experience, allowing them to track their progress and receive feedback on their performance.

The application can be easily integrated with existing educational systems, making it an efficient and effective way for educators to deliver outcome-based education and improve student learning outcomes.

In summary, Web-based Application is a powerful tool for educators and students. It provides an efficient and effective way to design and deliver curriculum that is aligned with desired learning outcomes, it allows for improved communication, and it provides students with a personalized learning experience. This application has the potential to greatly improve student achievement and engagement.

1.1 Vision Statement

"Our vision for our university department is to strive for educational excellence by embracing cutting-edge technology and innovative pedagogy. We aim to achieve this by implementing an outcome-based, web-based application that will not only enhance the personalization, engagement and progress tracking for our students, but also empower our educators to deliver education in a more effective and efficient manner. We believe that by providing our students with a personalized learning experience, we can empower them to reach their full potential and achieve their academic goals. Furthermore, by providing our educators with the tools and resources they need to deliver high-quality education, we can foster an environment of

continuous improvement and innovation. Together, we can create a university department that is truly dedicated to student success."

1.2 Related System Analysis/Literature Review

Application Name	Weakness	Proposed Project Solution
Campus Management System (CMS)	CMS included the student and courses management system, but it did not deal with the outcome-based education.	Our system OBE will solve the problem by defining the Program outcomes, and those program outcomes will be mapped with the course outcomes.

1.3 Project Deliverables

Functional requirements document: This document outlines the specific functionality that the OBE Web-based Application will provide, including user roles and access levels, navigation, and user interface design.

Technical design document: This document describes the technical architecture of the OBE Web-based Application, including the technologies that will be used to build it and the data structure.

Test plans and test cases: This deliverable is a set of test plans and test cases that will be used to test the OBE Web-based Application. It will ensure that the application meets the functional and technical requirements.

Deployment plan: This document outlines the steps that will be taken to deploy the OBE Web-based Application to the production environment. It will include details on how to install, configure and test the application.

OBE Web-based Application: The final deliverable of the project will be the OBE Web-based Application itself. It will be fully functional and tested, ready for use by educators and students.

Maintenance and support: It will be provided for a certain period after the deployment of the OBE Web-based Application, to ensure that the application is running smoothly and to address any issues that may arise.

1.4 System Limitations/Constraints

Implementation challenges: OBE can be challenging to implement in practice, and a web-based application may add an additional layer of complexity. It may require significant changes to traditional teaching methods and can be difficult for educators to adapt to.

LI-1: **Limited accessibility:** Some students and educators may not have access to the technology required to use the OBE Web-based Application, such as a computer or internet access. This could lead to disparities in education and limit the effectiveness of the system.

LI-2: **Limited scalability:** The OBE Web-based Application may not be able to handle large numbers of students and educators, which could limit its scalability and impact its overall performance.

LI-3: **Data security:** Web-based applications are vulnerable to cyber-attacks and data breaches, which could compromise the personal information of students and educators. It is crucial to ensure that the OBE Web-based Application has robust security measures in place to protect user data.

LI-4: **Limited personalization:** The OBE Web-based Application may not be able to provide a personalized learning experience for each student, as it may be difficult to tailor the content to meet the unique needs and learning styles of each individual.

LI-5: **Limited support:** OBE Web-based Application may not have adequate support and maintenance, which could lead to technical issues and limit the overall effectiveness of the system.

LI-6: **Cost:** Developing and implementing an OBE Web-based Application can be a costly endeavor, and it may require significant resources, including financial and human resources.

LI-7: **Time:** Developing and Implementing a web-based application can take time and effort, and it can be a long-term project which may not be feasible for every education institution to implement.

1.5 Tools and Technologies

The MEAN stack is a popular stack used for building web-based applications, and it can be used for the development of an Outcome-Based Education (OBE) Web-based Application as well. The MEAN stack stands for MongoDB, Express.js, AngularJS, and Node.js.

	Tools	Version	Rationale
Tools And Technologies	WebStorm	2023.1	IDE
	VS Code	2015	Source-code editor
	Draw.io	CSC 6	Design Work
	Technology	**Version**	**Rationale**
	HTML	5	Front-end Development
	Bootstrap	5	Front-end Development
	Angular CLI	15.1.2	Front-end Development
	Express	4.18.2	Back-end Development
	Node	16.14.0	Back-end Development
	MongoDB	Cloud-Based	Database
	NPM	8.3.1	Package manager

1.6 Relevance to Course Modules

The Outcome-based Education (OBE) web-based application project is related to various courses studied during a Bachelor of Science in Information Technology (BSIT) program in several ways:

1. The project involves the use of web development technologies, such as HTML, CSS, and JavaScript, which are covered in web development courses.
2. The project also utilizes a MEAN stack (MongoDB, ExpressJS, AngularJS, and NodeJS) for the development of the application, which we Are learning during building our Project.
3. The project involves the use of software development methodologies such as Agile, which is a topic covered in software engineering courses.
4. The project also involves the use of version control software like Git, which is a topic covered in software engineering and web development courses.
5. The project also involves the use of an Integrated Development Environment (IDE) like VSCode for the development of the application, which is a topic covered in software engineering and web development courses.
6. The project also involves the use of project management tools like Jira, which is a topic covered in project management courses.
7. The project also involves the use of Data modeling techniques and database management system, which is a topic covered in database management courses.
8. The project also involves the use of security measures to protect the data, which is a topic covered in cybersecurity and network security courses.

9. The project also involves the use of testing and quality assurance techniques to ensure the functionality and usability of the application, which is a topic covered in software engineering and quality assurance courses.

Overall, the OBE web-based application project is an excellent opportunity for BSIT students to apply the knowledge and skills they have acquired in various courses to a real-world project.

1.7 Problem Statement

There are many challenges faced by both educators and students in the education system. These include difficulties in tracking student progress, low engagement levels, and poor academic outcomes. The application will provide a comprehensive and user-friendly platform that facilitates personalized learning, engagement, and progress tracking. It will empower educators to deliver education more effectively and efficiently and enable students to achieve their academic goals and reach their full potential. The system will offer an easy and convenient way for educators to manage and track student progress, leading to increased engagement and improved academic outcomes. Additionally, it will provide students with access to the resources and tools necessary for success in their education. The overall goal of this project is to empower both educators and students to achieve their goals and reach their full potential in the education system through a web-based application for Outcome-Based Education.

1.8 Problem Solution

A web-based application for Outcome-Based Education is our suggested response to the present issues the educational system is facing. For both instructors and students, this program is intended to offer a thorough and user-friendly platform that supports individualized learning, engagement, and progress monitoring. The system will give teachers the tools they need to educate children more effectively and efficiently while also enabling them to fulfil their full potential as learners. The program will provide instructors with a simple and practical tool to oversee and monitor student development, increasing engagement and producing better academic results.

The system will also include real-time reporting and data analytics capabilities, giving teachers insightful information on the development and performance of their students. This application's capacity to adjust to the specific demands of each learner is one of its primary strengths. Each student will receive a tailored curriculum that takes into account their particular talents and limitations thanks to data-driven algorithms that will customize their learning experience. Additionally, this application will be used from any device with an internet connection, giving flexibility for both teachers and students to use the system from any

location. Users will find it simple to access the features and tools they require thanks to its responsive user interface and simple navigation.

1.9 Objectives of the Proposed System

The objectives of the proposed web-based application for Outcome-Based Education (OBE) are as follows:

OB-1: *To offer instructors and students a thorough and user-friendly platform for individualized learning, engagement, and progress monitoring.*

OB-2: *To provide educators with the instruments and materials they need in order to efficiently manage and monitor student development, which will enhance engagement and improve academic results.*

OB-3: *To enable students to achieve their academic goals and reach their full potential through access to personalized curriculum and resources.*

OB-4: *To offer real-time data analytics and reporting capabilities, providing educators with valuable insights into student performance and progress.*

OB-5: *To adapt to individual student needs by using data-driven algorithms to personalize the learning experience for each student.*

OB-6: *To make the application accessible from any device with an internet connection, providing flexibility for both educators and students.*

OB-7: *To design the application with a responsive user interface and easy navigation, making it easy for users to access the features and tools they need.*

OB-8: *To improve the education system by addressing the current challenges faced by educators and students.*

OB-9: *To provide a platform that can be easily integrated with existing education systems and processes.*

OB-10: *To provide a web-based solution that can be easily scaled and updated as needed.*

OB-11: *To improve the overall student learning experience and academic outcomes.*

1.10 Scope

The scope of the proposed Outcome-based Education (OBE) web-based application project is to provide a comprehensive solution for universities and colleges to implement the OBE system effectively. The main functionalities of the proposed project include:

1. Creation and management of outcome statements for different academic programs.
2. Alignment of course objectives, learning activities and assessments with the outcome statements.
3. Tracking and monitoring of student performance against the outcome statements.
4. Generation of reports on student performance and program outcomes.
5. Student and faculty portal for easy access to the outcome statements and performance reports
6. The ability to customize the application based on different educational institutions' requirements.
7. Secure data management and accessibility controls to ensure data privacy and security.
8. Mobile-friendly interface for ease of access
9. User-friendly navigation and interface
10. The ability to handle large data sets.
11. Scalable infrastructure to support future expansion.
12. Technical support and maintenance services post-deployment.

1.11 Modules

1.11.1 Module 1: Login Module

The login module is an essential component of the outcome-based education web-based application. It allows users to access the system by providing their credentials, such as their username and password. The module's main features include:

- **User authentication:** The login module verifies the user's identity by checking the provided credentials against the information stored in the system's database.
- **Secure password storage:** The login module stores user passwords in an encrypted format to ensure the security of users' personal information.
- **Logout feature:** The login module also includes a logout feature that allows users to end their session and log out of the system.

1.11.2 Module 2: Faculty

The Modules’ main features includes:

- **Add faculty:** User will be able to add faculty.

- **Update Faculty:** User will be able to Update existing record of faculty
- **Delete Faculty:** User will be able to Delete existing record of faculty.
- **View Faculty:** User will be able to View existing record of faculty.

1.11.3 Module 3: Curriculum

The Modules' main features includes:

- **Batches:** User will be able to Create, update, and delete Batches
- **Terms:** User will be able to Create, update, and delete Terms
- **Courses:** User will be able to Create, update, and delete Courses and add course outcome (CO)
    - **Course Outcome:** Add CO providing relevant information in form regarding course also update, delete and view CO
- **CO mapping with PO and PSO:** User will map CO with PO and save the map.

1.11.4 Module 4: Assessment

The Modules' main features includes:

- **Direct Assessment:** user will add assessment by selecting the appropriate curriculum, term and course. The assessment will be updated and deleted and will be viewed by user.

1.11.5 Module 5: Attainments

- **Data Import:** The user may download templates, import data, and examine assessment results by choosing the proper curriculum, term, and course.
- **Survey response:** The user will be able to download the template, import data, and evaluate the result about indirect assessment by choosing the relevant curriculum, term, and course.
- **CO Attainment:** By choosing relevant curriculum, term, and course the use will calculate the CO attainment by setting the maximum or minimum threshold value.

2 Requirement Analysis

Requirement analysis is the process of identifying and understanding the needs and constraints of a system to develop a solution that effectively meets those needs. It is an important step in the systems development life cycle and involves gathering and analyzing information from a variety of sources, including stakeholders, users, and subject matter experts. The goal of requirement analysis is to clearly define and document the functional and non-functional requirements of the system to ensure that the final solution meets the needs of the users and is aligned with the overall goals and objectives of the organization. In this process, various techniques like interviews, questionnaires, and observation may be used to collect data and information. The requirement analysis process can be iterative and may include the use of modeling techniques such as use case diagrams, event-response tables, and business rules to represent the requirements in a clear and concise manner.

2.1 User classes and characteristics

In the system, there are two main user classes:

1. **Administrators:** They are responsible for managing the system and have the highest level of access. They are responsible for creating and managing course content, grading assessments, and providing feedback to students.
2. **Faculty members:** They are responsible for creating and managing course content, grading assessments, and providing feedback to students.

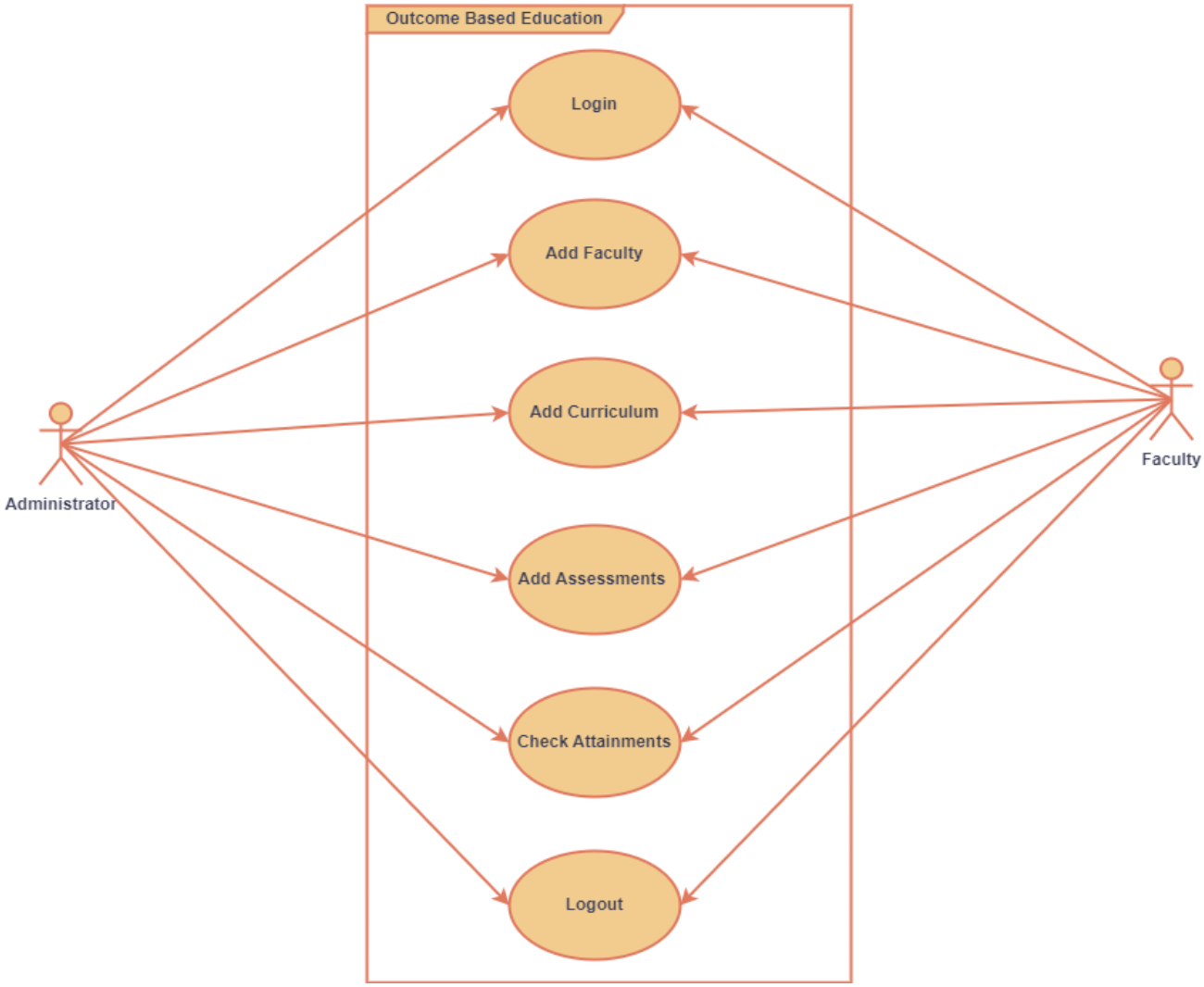
The characteristics of these user classes include:

1. **Administrators:** experienced with system management, able to handle sensitive information, familiar with technology.
2. **Faculty members:** experienced in their field of study, able to create and evaluate assessments, familiar with technology.

2.2 Requirement Identifying Technique

By understanding the needs and goals of the system, a clear and concise plan for development can be created, ensuring that the final product meets the needs of the end-users. In this project, we will be utilizing a combination of these techniques to identify the requirements for our web-

based outcome-based education system. This will include identifying the different user classes and their characteristics, creating use case diagrams to understand the system's functionality, and using event response tables to anticipate and plan for potential system events. Additionally, functional requirements will be defined to ensure that the system is able to meet the specific needs of our target user.



(Use Case Diagram)

Use Case ID:	UC-1
Use Case Name:	Login
Actors:	User (Admin or Faculty)
Description:	a. User opens the Outcome-Based Education System on their device. b. The system prompts the user to enter their username and password. c. User enters their credentials. d. The system verifies the entered credentials with the database. e. If the entered credentials are correct, the system grants the user access to the system. f. If the entered credentials are incorrect, the system displays an error message and denies the user access. g. The system records the successful or unsuccessful login attempts in the database.
Trigger:	The user clicks on the login button on the home page.
Preconditions:	a. The user has access to a device with a web browser. b. The user is connected to the internet. c. The system is accessible and functioning properly.
Postconditions:	a. The user is logged into the system. b. The user has access to their dashboard and can perform the tasks associated with their role (faculty or student). c. The system records the login event for audit purposes.
Flow of Events:	a. The user enters their username and password on the login page. b. The system verifies the username and password against the database. c. If the username and password are incorrect, the system displays an error message, and the user is unable to access the system. d. If the username and password are correct, the system grants access to the user and displays the user's dashboard.
Exceptions:	In case of forgotten password, the user cannot use the forgot password feature to reset it.
Business Rules	a. Users are required to enter a valid username and password to access the system. b. The username and password must be verified against the system's database. c. If the username and password are incorrect, the user will be unable to access the system. d. If the username and password are correct, the user will be granted access to the system.
Assumptions:	a. The system has a database to store user information and authentication details. b. The user has already registered and has a valid username and password.

Use Case ID:	UC-2
Use Case Name:	Add Faculty
Actors:	User (Admin or Faculty)
Description:	a. The faculty use case allows faculty members to log in to the Outcome Based Education system and access their specific account. b. Once logged in, they can view and manage course information, curriculum details, assessments, and student performance.
Trigger:	• The user clicks on the "Add Faculty" button in the faculty module.
Preconditions:	a. The admin must have a valid account in the Outcome Based Education system. b. The admin must have the correct login credentials (username and password).
Postconditions:	a. The new Faculty's information is stored in the database and is available for future use. b. The admin can now assign courses and assessments to the new Faculty.
Flow of Events:	a. The admin selects the option to add a new Faculty from the main menu. b. The system displays the Add Faculty form. c. The admin inputs the necessary information, such as the Faculty's name, email address, and qualifications. d. The admin reviews the inputted information and confirms it is correct. e. The system checks if the email address already exists in the database. f. If the email address already exists, the system displays an error message and the admin must input a different email address. g. If the email address is unique, the system stores the new Faculty's information in the database. h. The system displays a confirmation message to the admin, indicating that the new Faculty has been added successfully. i. The admin has the option to return to the main menu or add another Faculty
Exceptions:	a. Invalid Input: If the user enters invalid information while adding a faculty, the system should prompt an error message and ask the user to re-enter the information. b. Duplicate Entry: If the user tries to add a faculty with the same name or ID as an existing faculty, the system should prompt an error message and ask the user to enter unique information. c. Database Error: If there is an error in the database while adding the faculty, the system should prompt an error message and ask the user to try again later. d. Network Failure: If the system is unable to connect to the database or any other network resource, it should prompt an error message and ask the user to try again later.
Business Rules	a. The email address must be unique and cannot be used by another Faculty. b. The Faculty's name and qualifications must be inputted correctly. c. Empty field are not allowed
Assumptions:	a. Admin has the necessary privileges to add a new faculty member to the system. b. The system has the capability to store and manage faculty member information.

Use Case ID:	UC-3
Use Case Name:	Add Curriculum
Actors:	User (Admin or Faculty)
Description:	The process of adding a new curriculum to the system in Outcome Based Education. The new curriculum will be added to the system and can be used by faculty members when creating courses
Trigger:	• The user clicks on the "Courses" button or "Curriculum" on Home Page.
Preconditions:	c. The admin must have a valid account in the Outcome Based Education system. d. The admin must have the correct login credentials (username and password).
Postconditions:	a. When use Trigger the button, it will lead to course module where user will add information of course and add CO of that course and the course data will be save to database
Flow of Events:	a. Admin will select the option "curriculum or courses" from the home page b. Button trigger redirect to courses page where admin will click add course button c. On Clicking button the new form will be pop-up d. Admin will add the information about course and then save the course e. After saving course CO button will be showed on saved course f. Admin click on CO button and add all information about CO. g. The new Curriculum will be created or saved to database
Exceptions:	a. Invalid Input: If the user enters invalid information while adding a course, the system should prompt an error message and ask the user to re-enter the information. b. Duplicate Entry: If the user tries to add a course with the same name or ID as an existing course, the system should prompt an error message and ask the user to enter unique information. c. Database Error: If there is an error in the database while adding the course, the system should prompt an error message and ask the user to try again later. d. Network Failure: If the system is unable to connect to the database or any other network resource, it should prompt an error message and ask the user to try again later.
Business Rules	a. Curriculum must have at least one course b. Without course CO will not be added c. Without curriculum other operation will not be done such as, adding assessment to course
Assumptions:	a. The user has the necessary permissions to add a new curriculum to the system. b. The user has entered the required information correctly.

Use Case ID:	UC-4
Use Case Name:	Add Assessment
Actors:	User (Admin or Faculty)
Description:	This use case describes the process of adding a new assessment to the curriculum in the Outcome-Based Education system. The administrator will be able to provide the details of the assessment such as its name, type, date, weightage, and the CO it is associated with.
Trigger:	• The user clicks on the "Assessment" button on Home Page.
Preconditions:	a. The administrator must be logged into the system. b. The Curriculum, Term and Course must be selected before adding Assessment.
Postconditions:	a. When user select the appropriate curriculum, term and course the add assessment button will be clickable b. The user will be allowed to add Assessment by clicking on assessment button. c. The admin will provide the relevant information of assessment
Flow of Events:	a. The system displays the add assessment form. b. The administrator enters the details of the assessment such as its name, type, date, weightage, and the course it is associated with. c. The system validates the entered information. d. If the information is valid, the system saves the assessment to the database and displays a confirmation message. e. If the information is not valid, the system displays an error message and prompts the administrator to correct the information.
Exceptions:	a. If the Curriculum, Term and Course does not select the Add Assessment button will not be Clickable
Business Rules	a. Assessment must have a name. b. Assessment must have type. c. Assessment must have a Question. d. Assessment must have a CO
Assumptions:	a. The administrator has the necessary permissions to add an assessment.

Use Case ID:	UC-5
Use Case Name:	Check Attainments
Actors:	User (Admin or Faculty)
Description:	The Check Attainments use case allows the user to view the attainments of Student over CO and Co over PEO
Trigger:	The user clicks on the “Attainment” button or “Data Import” on Home Page.
Preconditions:	a. The user must have valid login credentials. b. Mapping or CO and PEO c. The student must have already taken the assessments for the course or curriculum. d. The assessments must have already been evaluated and scores entered into the system.
Postconditions:	a. The user has access to the attainments of student and CO in the selected course or curriculum. b. The user can view and Downland Template of the attainments.
Flow of Events:	a. The user logs into the system with their credentials. b. The user selects the "Check Attainments" option from the main menu. c. The system displays a list of courses or curriculums available for the user to view attainments. d. The user selects a specific course or curriculum. e. The system displays the list of students and their attainments for the selected course or curriculum. f. The user may choose to view additional details for each student, such as their individual scores for each assessment. g. The user may choose to print or export the attainments report.
Exceptions:	a. If the Curriculum, Term and Course does not select the attainment will not be viewed or downloaded
Business Rules	a. Appropriate Curriculum, term and Course must be Selected for all operation such as CO Attainment, CO Over PEO Attainments result of Students. b. Benchmarks have to be added for CO attainment calculation
Assumptions:	a. The system has the capability to store and retrieve student attainments. b. The assessments have already been evaluated and scores entered the system. c. The user has the necessary permissions to access and view attainments.

2.3 Functional Requirements

Functional requirements are the specific functions and features that a system must have in order to fulfill its intended purpose. In other words, functional requirements describe what the system must do, rather than how it will do it. In the context of an outcome-based education system, functional requirements are critical for ensuring that the system meets the needs of its stakeholders and achieves its goals. This includes ensuring that the system is user-friendly, secure, and able to perform the necessary functions for managing curriculum, assessment, and student achievement. A well-defined set of functional requirements is essential for guiding the design and development of the system, and for communicating the expected outcomes and capabilities to stakeholders.

2.3.1 Functional Requirement X

Identifier	FR-1
Title	User Authentication
Requirement	**From User Perspective** The User shall be able to login to the system using a valid username and password. The login process should be quick and secure, and the user should be able to access the system within 5 seconds of entering the correct login credentials. **From System Perspective** Upon receipt of a login request from the user, the system shall verify the user's username and password against the database. If the login credentials are valid, the system shall allow the user to access the system and display the user's homepage. If the login credentials are invalid, the system shall display an error message indicating that the login has failed and prompt the user to try again.
Source	Web Application
Rationale	It gives the permission to only authorized user
Business Rule (if required)	**User Identification:** The system must have a mechanism for identifying users, such as a username and password, to ensure secure access. **Authentication:** The system must have a secure authentication mechanism to verify the identity of the user before granting access to the system. **Authorization:** The system must have a mechanism for determining

	which users have access to which features and functionality within the system.
Dependencies	Availability of a secure and reliable authentication mechanism such as user ID and password.
Priority	High

Identifier	FR-2
Title	Add Faculty
Requirement	**User Perspective** The admin shall be able to add new faculty members to the system by providing their personal and professional information such as name, email, designation, and department. The faculty member's information should be added within 2 minutes with high accuracy. **System Perspective** Upon the trigger of the "Add Faculty" feature, the system shall collect the required information from the admin, validate the entered data, and store it in the database. The system shall display a confirmation message after successfully adding the faculty member's information to the database. In case of any invalid data, the system shall prompt an error message to the admin for correction.
Source	Web Application
Rationale	It gives the permission to only authorized user
Business Rule (if required)	The user shall be able to add a new faculty member to the system by providing the required information such as name, email, department, and role. The system shall validate the provided information and ensure that it meets the necessary constraints and requirements. If the information is valid, the system shall add the new faculty member to the database.
Dependencies	a. User authorization to add faculty. b. Faculty information must be complete and accurate.
Priority	High

Identifier	FR-3
Title	Add Curriculum
Requirement	**From user perspective** The administrator shall be able to add a new curriculum/ add course or CO to the system by providing the necessary information such as the curriculum name, description, and subject areas. **From system perspective** Upon receiving a request to add a curriculum / add course or CO, the system shall validate the input data, store the information in the database, and display a success message to the administrator.
Source	Web Application
Rationale	It gives the permission to only authorized user
Business Rule (if required)	The business rule for adding the curriculum in the outcome-based education system states that only authorized users, such as administrators, can add or modify curricula. The curricula must meet certain criteria, such as being up-to-date and aligned with the desired educational outcomes. The system must also ensure that each curriculum is unique and cannot be duplicated. The addition or modification of the curriculum must be recorded in the system for auditing purposes.
Dependencies	a. User authorization to add curriculum. b. Course information must be complete and accurate. c. Information of CO must also be complete and accurate with respect to course
Priority	High

Identifier	FR-4
Title	Add Assessment
Requirement	**User perspective** The administrator shall be able to add assessments to specific courses with the required details such as type, weightage, and Questions and its relevant CO. **System perspective** On the trigger event of the administrator requesting to add an assessment, the system shall validate the course information and collect the assessment details. The system shall then store the assessment information in the database and display a confirmation message to the administrator.
Source	Web Application
Rationale	It gives the permission to only authorized user
Business Rule (if required)	The business rule for adding assessment requires that only authorized users, such as teachers or administrators, have the capability to create new assessments. The system must ensure that the assessments are aligned with the curriculum and CO. The system must also provide error messages or prevent invalid inputs, such as incorrect date format or negative point values.
Dependencies	The dependencies of adding an assessment would include the availability of the curriculum the course, and CO to which the assessment is being added, as well as the presence of any necessary faculty or administrative permissions to add the assessment. The specific details of these dependencies would depend on the policies and processes of the institution in question.
Priority	High

Identifier	FR-5
Title	Check Attainments
Requirement	**User perspective** The user shall be able to check the attainments of students for a specific assessment also check attainment of CO over PEO [qualifying conditions, response time, or quality statement]. **System perspective** Upon the request of the user, the system shall provide the attainments of

	students for a specific assessment and CO attainment over PEO [expected system response].
Source	Web Application
Rationale	It gives the permission to only authorized user
Business Rule (if required)	The business rule for adding assessment requires that only authorized users, such as teachers or administrators, have the capability to create new assessments. The system must ensure that the assessments are aligned with the curriculum and CO. The system must also provide error messages or prevent invalid inputs, such as incorrect date format or negative point values.
Dependencies	The dependencies of adding an assessment would include the availability of the curriculum the course, and CO to which the assessment is being added, as well as the presence of any necessary faculty or administrative permissions to add the assessment. The specific details of these dependencies would depend on the policies and processes of the institution in question.
Priority	High

2.4 Non-Functional Requirements

Non-functional requirements (NFRs) are those requirements that specify the quality attributes or constraints of the system, such as performance, usability, security, and compatibility. NFRs define the performance and behavior characteristics of a system and can have a significant impact on the overall success of the project. They define the operational and environmental constraints under which the system must function and set the standards for its performance, reliability, and user experience. All these NFRs explained below.

2.4.1 Usability

Usability refers to the ease of use and learnability of a product, system, or software. In the context of outcome-based education, the usability of the system would be evaluated based on factors and usage:

FAC / USE-1: *Intuitive navigation and interface design, allowing users to easily find what they are looking for and perform tasks.*

FAC / USE -2: *Clear and concise instructions and prompts, reducing the likelihood of confusion or frustration.*

FAC / USE -3: *All the data will be fetched automatically when the user login to system including all faculty details, curriculum details and Assessments detail.*

FAC / USE -4: *Attainments data will be calculated as per needs of the user.*

FAC / USE -5: *Efficient and effective task completion, allowing users to achieve their goals with a minimum of time and effort.*

2.4.2 Performance

The performance of system will depend on various factors such as the complexity of the system, the amount of data being processed, the network infrastructure, the hardware specifications, and the user load. The Application is developed using MEAN stack so it's not reload the page on every click so its render any page in 2,3 second when doing asynchronous task it will also it always consume the same time but sometimes it will depend on network Infrastructure as above explained.

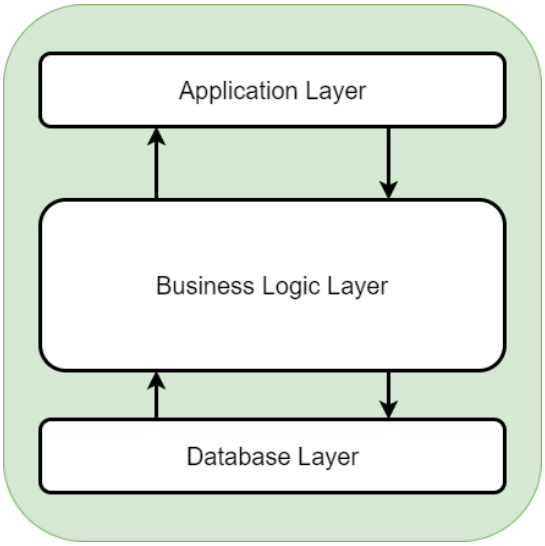
2.4.3 Security

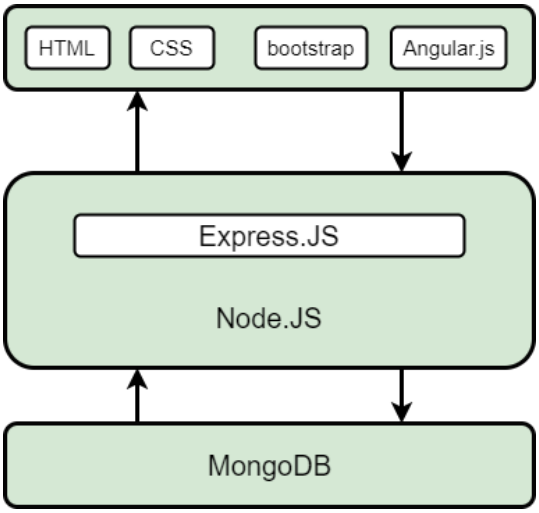
System should ensure the protection of sensitive information and resources, such as personal information, assessment results, and educational materials. The security measures should include user authentication, access control, data encryption, backup and recovery, and regular security audits. The system should be designed to prevent unauthorized access, hacking attempts, and other security threats. The consequences of security breaches, such as data loss or theft, should be clearly defined and a plan for responding to security incidents should be in place.

3 Design and Architecture

The following parts of Software Design Description (SDD) report should be included in this chapter.

3.1 Architectural Design

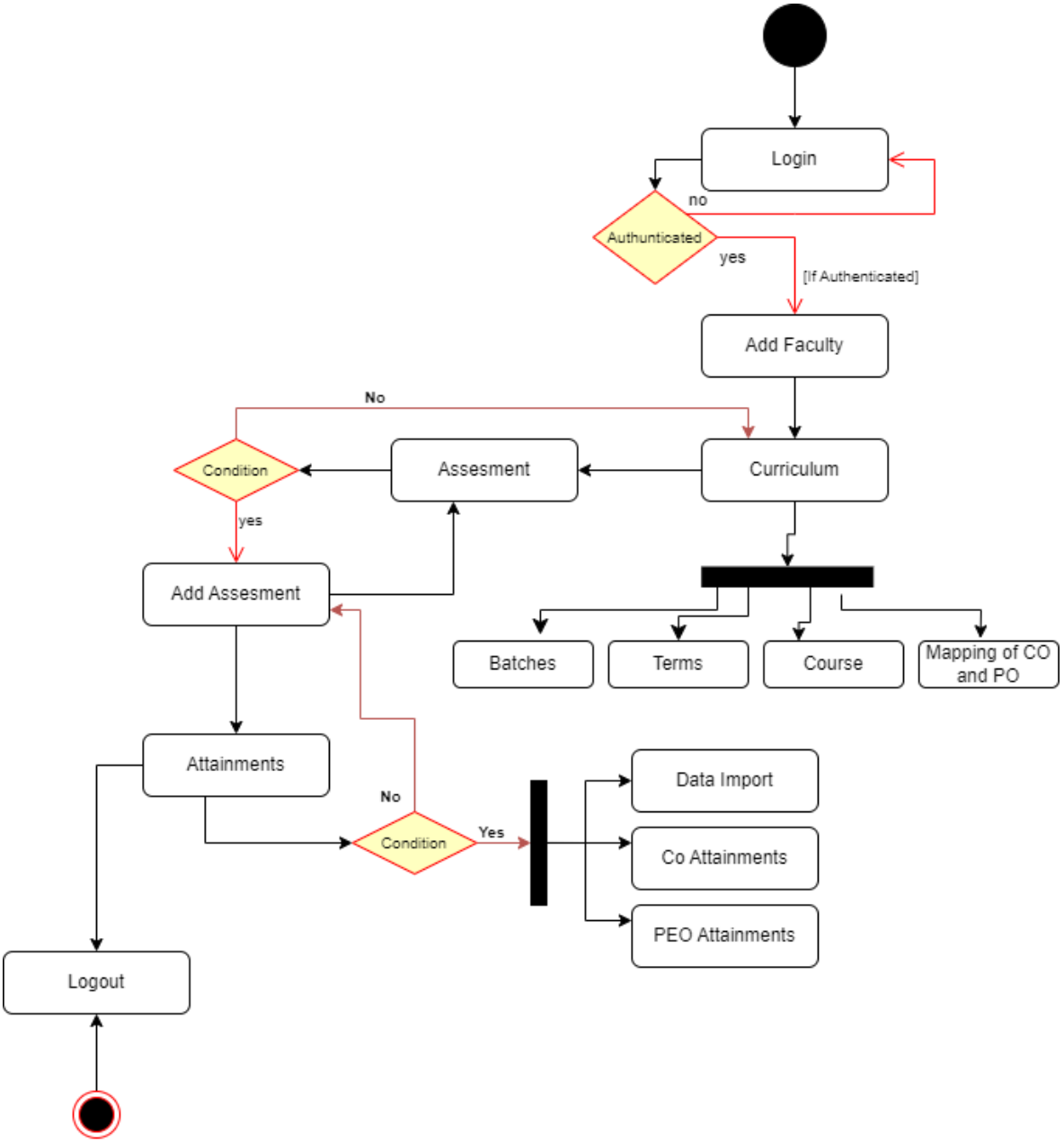




(Architecture Diagram)

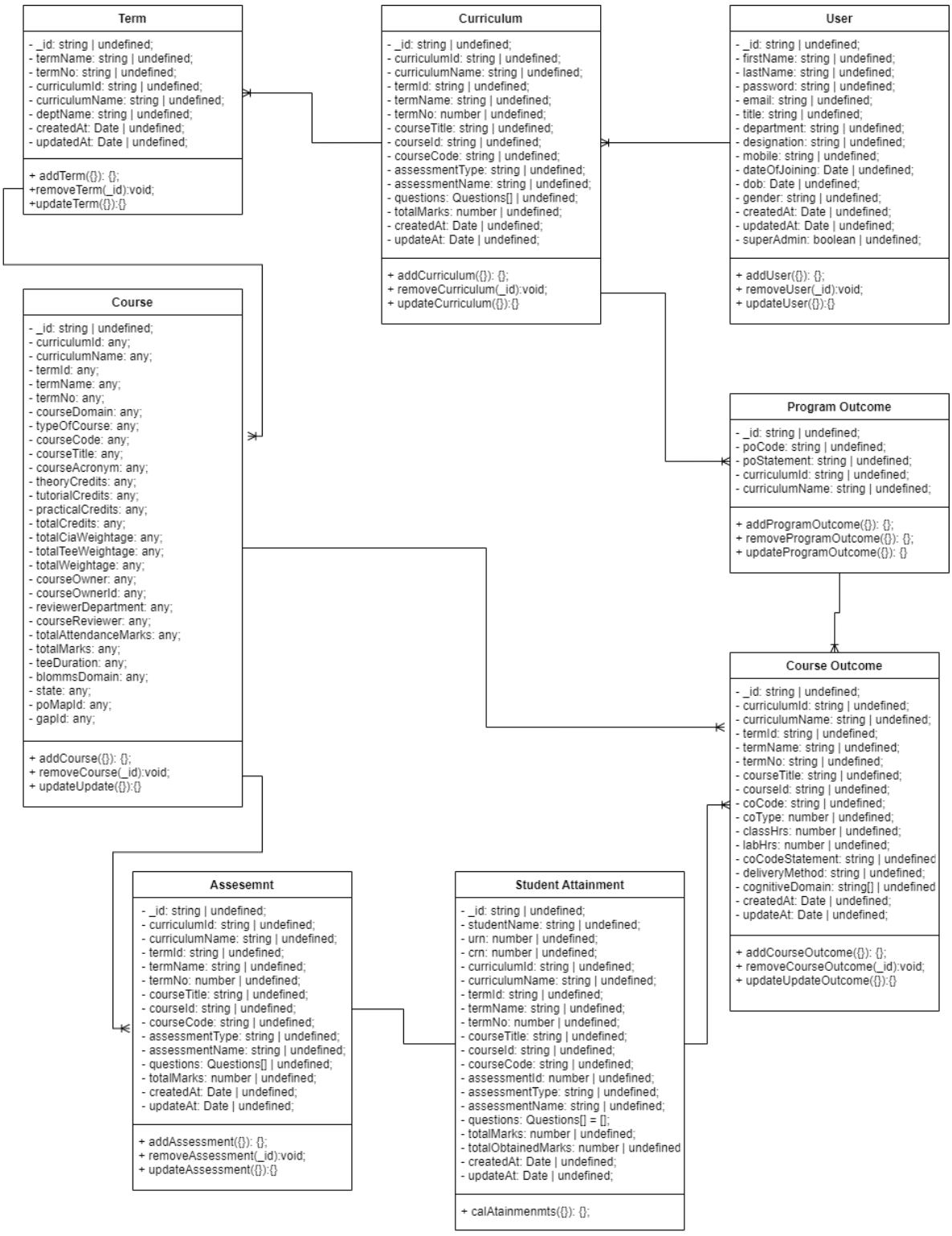
3.2 Design Models

3.2.1 Activity Diagram



(Activity Diagram)

3.2.2 Class Diagram



(Class Diagram)

3.3 Data Design

The major data or system entities can be stored as collections, which is equivalent to tables in relational databases. The collections can be organized into databases, which are used to store related collections.

Each document within a collection represents a single entity and contains a set of key-value pairs, which can represent the attributes of the entity. For example, if the entity represents a student, the document may contain attributes such as the student's name, enrollment date, and scores on assessments.

3.3.1 Data Storage Item:

MongoDB: A NoSQL database used for storing and processing large amounts of unstructured data.

MongoDB Atlas: A cloud-based database service offered by MongoDB that provides scalability, security, and high availability.

3.3.2 Data Dictionary

Entities	Data Type	Description

Assessment	Object	Assessment {
		_id: string | undefined;
		curriculumId: string | undefined
		curriculumName: string | undefined;
		termId: string | undefined;
		termName: string | undefined;
		termNo: number | undefined;
		courseTitle: string | undefined;
		courseId: string | undefined;
		courseCode: string | undefined;
		assessmentType: string | undefined;
		assessmentName: string | undefined;
		questions: Questions[] | undefined;
		totalMarks: number | undefined;
		createdAt: Date | undefined;
		updateAt: Date | undefined;
		}

Course	Object	Course { _id: string | undefined;

		curriculumId: any; curriculumName: any; termId: any; termName: any; termNo: any; courseDomain: any; typeOfCourse: any; courseCode: any; courseTitle: any; courseAcronym: any; theoryCredits: any; tutorialCredits: any; practicalCredits: any; totalCredits: any; totalCiaWeightage: any; totalTeeWeightage: any; totalWeightage: any; ciaPassingMarks: any; prerequisiteCourses: any; courseOwner: any; courseOwnerId: any; reviewerDepartment: any; courseReviewer: any; lastDateToReview: any; totalCourseConatactHours: any; totalCiaMarks: any; totalMidTermMarks: any; totalTeeMarks: any; totalAttendanceMarks: any; totalMarks: any; teeDuration: any; blommsDomain: any; state: any; poMapId: any; gapId: any; }
Curriculum	Object	Curriculum { _id: string | undefined; curriculumName: string | undefined; curriculumOwner: string | undefined; curriculumOwnerId: string | undefined; deptName: string | undefined; credits: number | undefined; state: boolean | undefined; minDuration: number | undefined; maxDuration: number | undefined;

		totalTerms: number | undefined; startYear: number | undefined; endYear: number | undefined; createdAt: Date | undefined; updatedAt: Date | undefined; terms: Term[] | undefined; }
CourseOutcomes	Object	CourseOutcomes { _id: string | undefined; curriculumId: string | undefined; curriculumName: string | undefined; termId: string | undefined; termName: string | undefined; termNo: string | undefined; courseTitle: string | undefined; courseId: string | undefined; coCode: string | undefined; coType: number | undefined; classHrs: number | undefined; labHrs: number | undefined; coCodeStatement: string | undefined; deliveryMethod: string | undefined; cognitiveDomain: string[] | undefined; createdAt: Date | undefined; updateAt: Date | undefined;}

StudentAttainments	Object	StudentAttainments {
		_id: string | undefined;
		studentName: string | undefined;
		urn: number | undefined;
		crn: number | undefined;
		curriculumId: string | undefined;
		curriculumName: string | undefined;
		termId: string | undefined;
		termName: string | undefined;
		termNo: number | undefined;
		courseTitle: string | undefined;
		courseId: string | undefined;
		courseCode: string | undefined;
		assessmentId: number | undefined;
		assessmentType: string | undefined;
		assessmentName: string | undefined;
		questions: Questions[] = [];
		totalMarks: number | undefined;
		totalObtainedMarks: number | undefined;
		createdAt: Date | undefined;
		updateAt: Date | undefined;

		}

Term	Object	Term {
		_id: string | undefined;
		termName: string | undefined;
		termNo: string | undefined;
		curriculumId: string | undefined;
		curriculumName: string | undefined;
		deptName: string | undefined;
		createdAt: Date | undefined;
		updatedAt: Date | undefined;
		}

User	Object	User {
		_id: string | undefined;
		firstName: string | undefined;
		lastName: string | undefined;
		password: string | undefined;
		email: string | undefined;
		title: string | undefined;
		department: string | undefined;
		designation: string | undefined;
		mobile: string | undefined;
		dateOfJoining: Date | undefined;
		dob: Date | undefined;
		gender: string | undefined;
		createdAt: Date | undefined;
		updatedAt: Date | undefined;
		superAdmin: boolean | undefined;
		}

4 Implementation

This chapter will discuss implementation details of the project.

4.1 Algorithm

calculateAttainemnt(stdRecords: StudentAttainments[])

Input: An array of student records (stdRecords) with each record containing student information including the questions attempted by the student

Output: A list of CO codes with the attainment percentage and level

Step 1: Create an object CO_Object with CO1 to CO6 as keys and default values count=0, maximumMarks=0, attainment="Not Applicable", attainmentLevel=0, attainmentType="Very Low", attaimentPercentage=0

Step 2: If the length of stdRecords is 0, return the values of CO_Object.

Step 3: Get the total number of questions (totalQuestions) from the student records.

Step 4: For each CO code in CO_CODE, calculate the maximum marks for each code and store it in CO_Object.

Step 5: Remove students from the student records who have attempted all the questions and obtained 'A' in all of them.

Step 6: For each student in the student records, calculate the attainment percentage for each CO code and increment the count for that CO code in CO_Object.

Step 7: For each CO code, if the count is not 0, calculate the attainment percentage and update the attainment, attainment level, and attainment type for that CO code in CO_Object.

Step 8: Return the values of CO_Object.

Algorithm 2 CO mapping With Po (Get POMap)	**Algorithm 2.1 similarTitles**
Input: Get PO in The Form	**Input:** Reinitialize the Form of CO PO mapping
Function getPOMap() **Step 1:** Make a GET request to the specified URL with headers **Step 2:** Upon success of the request, store the returned poMap in "selectedPO" and call "reInitialiseForm" function **Step 3:** Upon failure of the request, do nothing End Function	**Step 1:** Create an empty form group object **Step 1:** For the given course outcome code, program outcome and strength, add the values to the form group object as key value pairs **Step 3:** Push the form group object to the form array 'POForm' End.

Algorithm 2.2 add to form and Save CO/PO Mapping

Input: Edit The CO/PO Mapping Table
Output: CO/PO mapping will be save to database

1. Declare a local variable **coFormGroup** of type **FormGroup**.
2. Set the value of **coFormGroup** to the value returned by **this.coMappingForm.get(coCode)** as **FormGroup**.
3. Check if the **coFormGroup** already contains the passed PO code **poCode** by calling **coFormGroup.contains(poCode)**.
4. If the **coFormGroup** contains the PO code **poCode**, then check if the strength is not equal to 0. a. If the strength is not equal to 0, then set the control for the PO code **poCode** by calling **coFormGroup.setControl(poCode, this.fb.control(Number(strength)))**. b. If the

strength is equal to 0, then remove the control for the PO code **poCode** by calling **coFormGroup.removeControl(poCode)**.

5. If the **coFormGroup** does not contain the PO code **poCode**, then add the control for the PO code **poCode** by calling **coFormGroup.addControl(poCode, this.fb.control(Number(strength)))**.

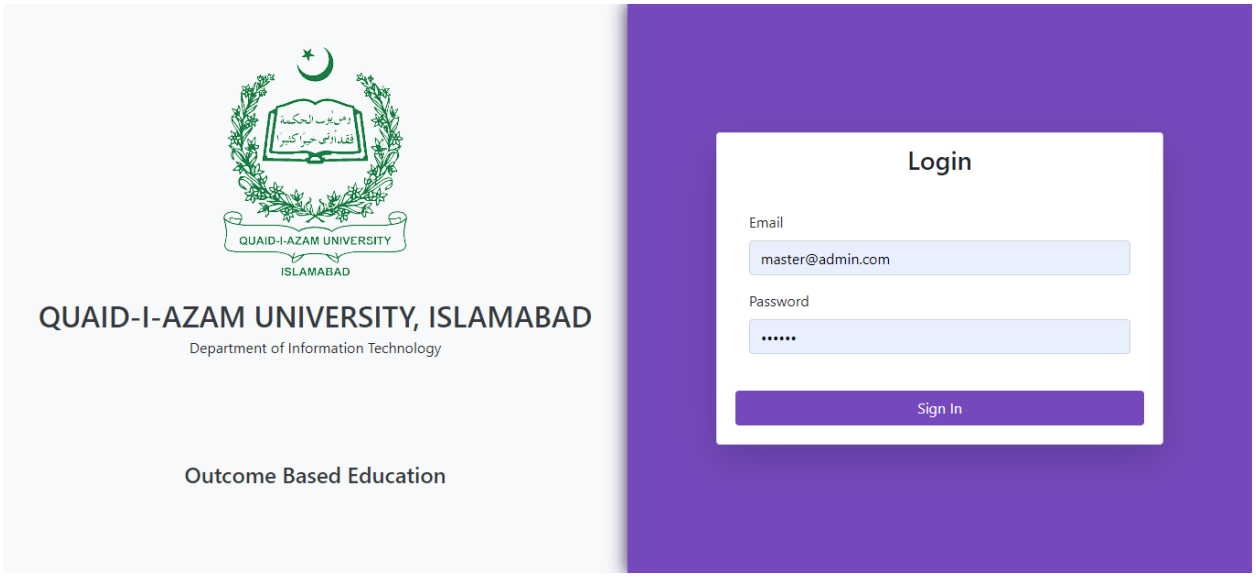
Pseudocode for the function **saveCOPOMapping**:

1. Declare a local variable **values** and set its value to an object that contains the values from **this.coMappingForm.value**.
2. Make a **post** HTTP request to the API endpoint **${environment.serverUrl}/co_po_mapping/add** with the **values** object and headers **this.dataService.httpHeaders** by calling **this.httpClient.post**.
3. Handle the response from the API by using the **.toPromise** method and call the **.then** function on it.
4. In the first function passed to **.then**, log the returned value to the console and display a success toast message using **this.toast.success("Co PO Mapping Save Successfully", "Success")**.
5. In the second function passed to **.then**, log the error to the console and display a warning toast message using **this.toast.warning("Something Went Wrong!! Please Try Again", "Error")**.

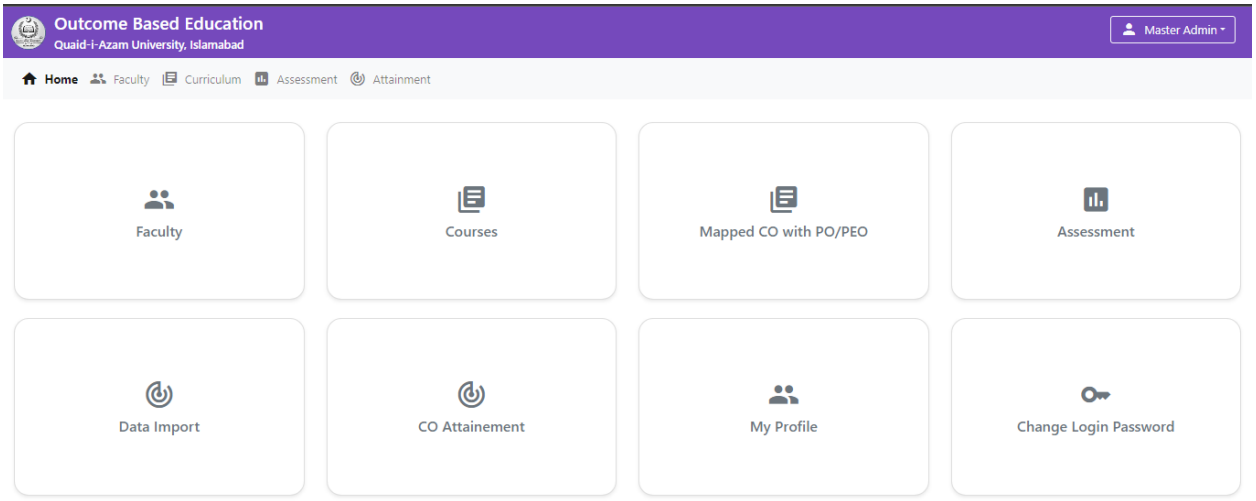
4.2 User Interface

This section shows the user interface of the application.
"Design is not just what it looks like and feel like. Design is how its work." ~ Steve Jobs

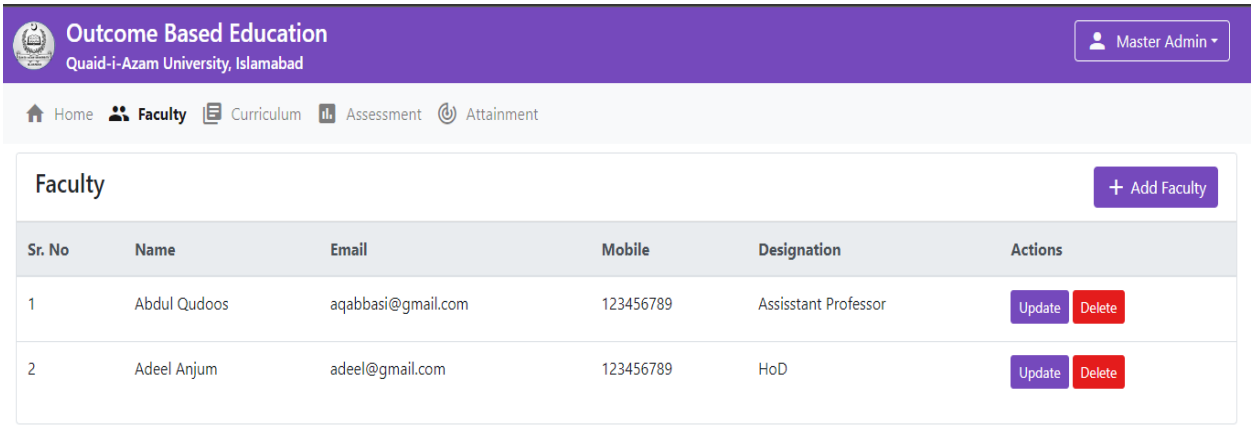
4.2.1 Login Screen



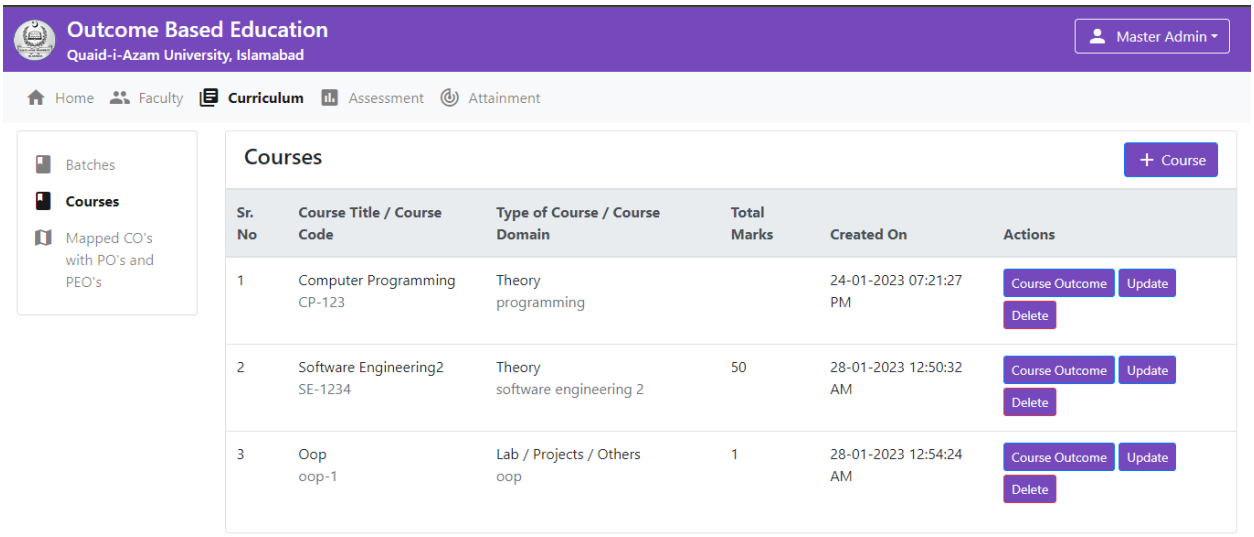
4.2.2 Home Page



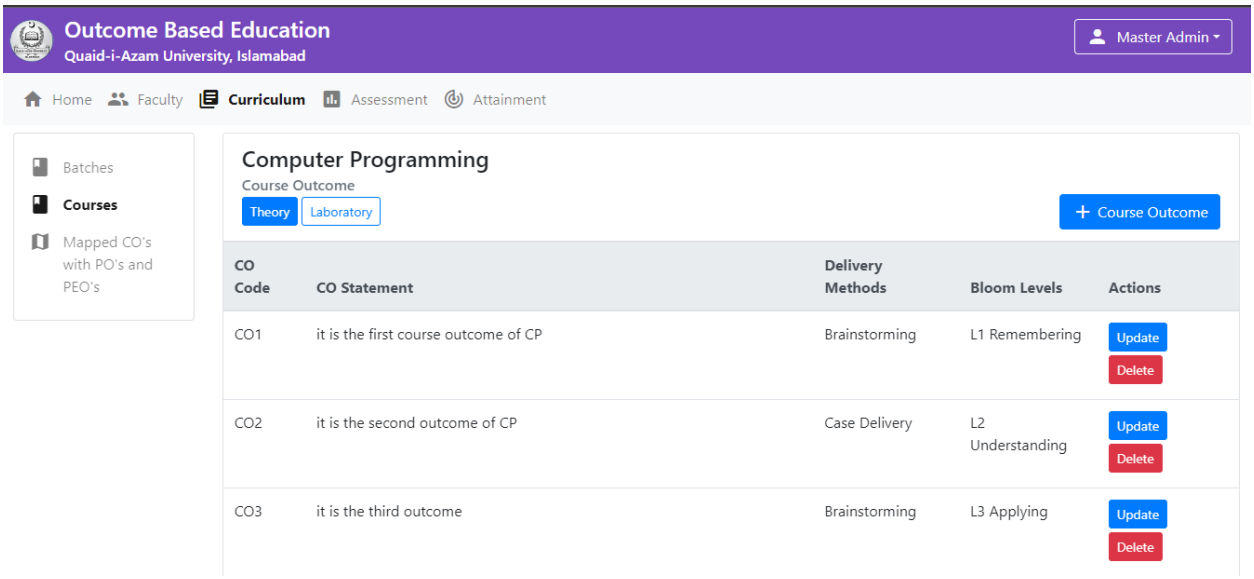
4.2.3 Faculty Page



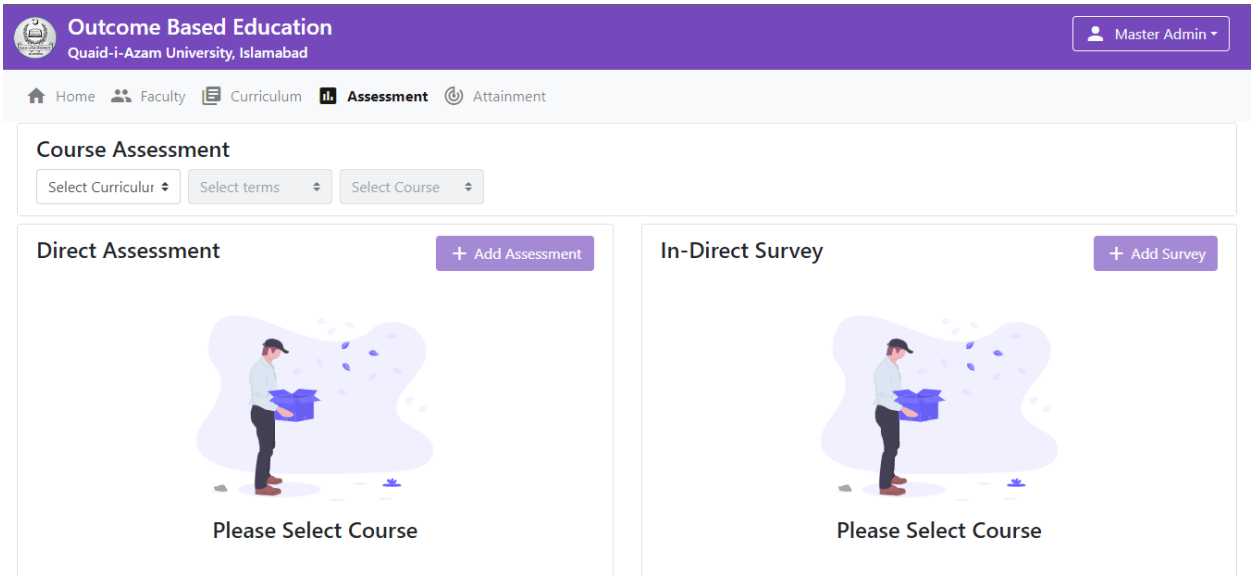
4.2.4 Curriculum Page



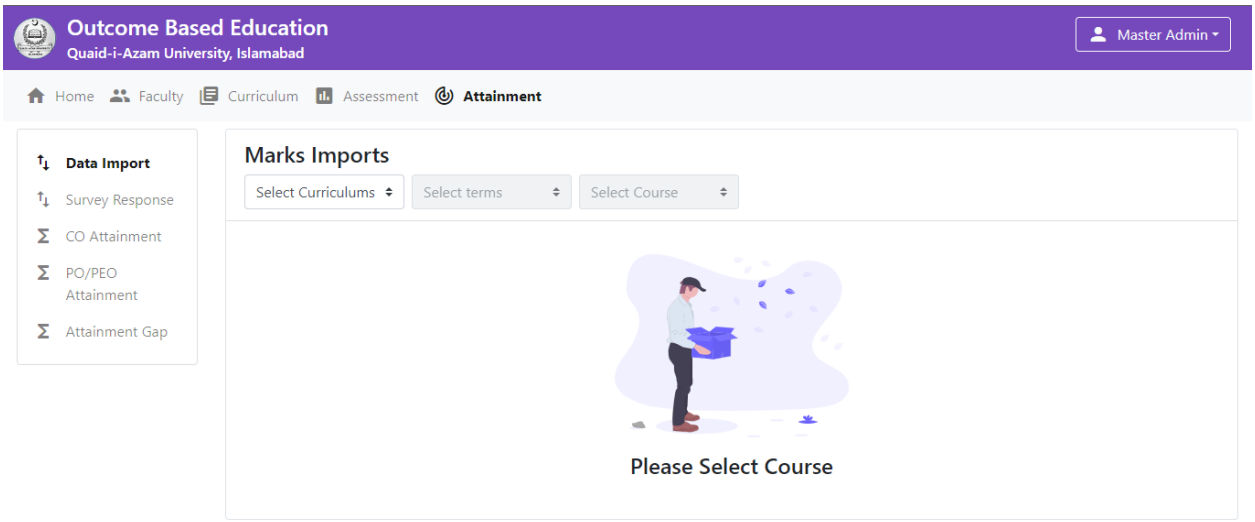
4.2.5 Course Outcome Page



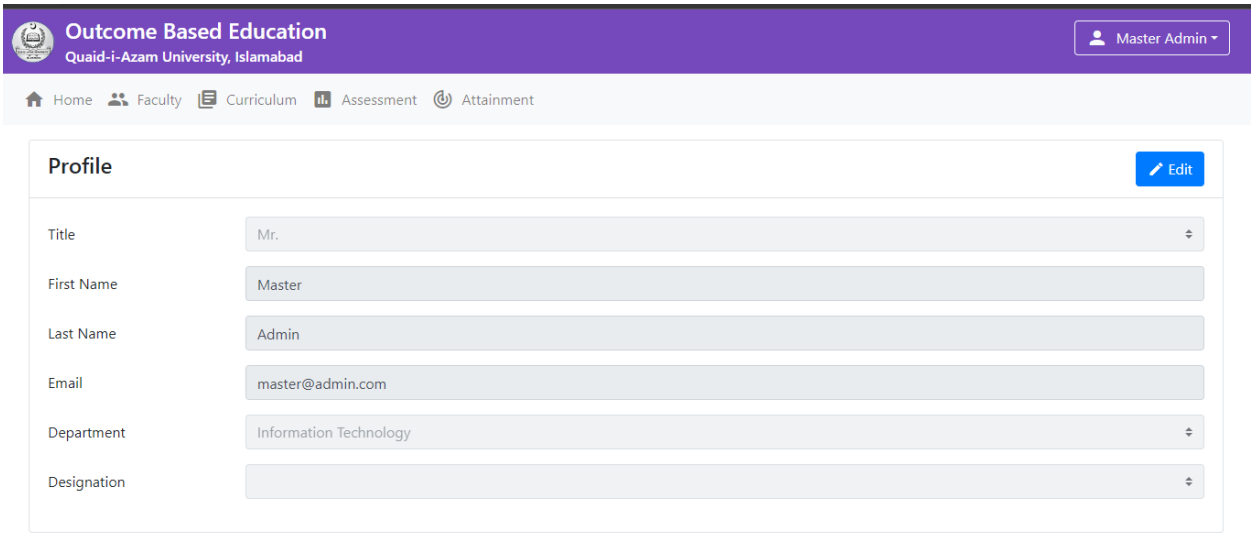
4.2.6 Assessment Page



4.2.7 Attainments Page



4.2.8 User Profile Page



5 Testing and Evaluation

This Chapter provide all the information about Testing and Evaluation of the application. It will monitor and test all the feature of over system is it works correctly.

5.1 Unit Testing

Unit Testing 1: Login with valid and invalid credentials.

Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of login to validate that it takes proper email	Email: master@admin.com	Validates the form	Pass
2	Check the email field of login to validate that it displays error message.	Email: abc.gmail.com	Highlights field and displays error message	Pass
3	Check the password Field that it takes correct password	Password: master	Validates the form	Pass
4	Check the Password field for incorrect Password	Password: Mister12	Display the Appropriate error on entering wrong password	Pass

Unit Testing 2: Adding Faculty with valid and invalid credentials.

Testing Objective: To ensure the faculty form is working correctly with valid and invalid credentials/inputs.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check all the field by adding Correct data in the field	Title: Dr First Name: Adeel Last Name: Anjum Mobile: 123456789 Email: adeel@gmail.com Password: auto generated Department: Information Technology Designation: HoD Gender: Male Data of Birth: 03/02/1996 Date of Joining: 03/02/2022	Add the faculty to the faculty module after valid Data	Pass
2	Check all the field by leaving any field empty in the form	Title: Dr First Name: Adeel Last Name: Anjum Mobile: 123456789 Email: Password: auto generated Department: Information Technology Designation: HoD Gender: Male Data of Birth: 03/02/1996 Date of Joining: 03/02/2022	Display the Error message	Pass
3	Entering the Same email in the field which is already added in the database	Email: adeel@gmail.com	Display the error message of already registered email	Pass

Unit Testing 3: Adding Batches with Entering data Into From.

Testing Objective: To ensure the Batches form is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result.	Result
1	Check all the field by adding Correct data in the field	Batch Name: MSc 2023-2025 Department: Information Technology Owner Name: Auto Total Credits:3 Total Terms:8	Add batches to the Batch module	Pass

		Start Year:2023 Ending Year:2027 Min Duration:8 Max Duration:12		
2	Check all the field by leaving any field empty in the form	Batch Name: Empty Min Duration: Empty Max Duration: Empty	Display the Error message	Pass

Unit Testing 4: Adding Term with Entering data Into From.

Testing Objective: To ensure the Term form is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result.	Result
1	Check all the field by adding Correct data in the field	TermName: Third TermNo:2 Department: Information Technology Owner Name: Auto Curriculum: Selected from List	Add batches to the Term module	Pass
2	Check all the field by leaving any field empty in the form	Term Name: Empty	Display the Error message	Pass

Unit Testing 5: Adding Course with Entering data Into From.

Testing Objective: To ensure the Course form is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result.	Result
1	Check all the field by adding Correct data in the field	Filled All the Fields	Add Course to the Curriculum Module	Pass

2	Check all the field by leaving any field empty in the form	Course Name: Empty	Display the Error message	Pass

Unit Testing 6: Adding Course outcome with Entering data Into From.

Testing Objective: To ensure the course outcome form is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result.	Result
1	Check all the field by adding Correct data in the field	Filled All the Fields	Add Course outcome to the Curriculum module	Pass
2	Check all the field by leaving any field empty in the form	Course Name: Empty	Form cannot be submitted. Submit button cannot be clicked	Pass

Unit Testing 7: Adding Assessment with Entering data Into From.

Testing Objective: To ensure the Assessment form is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result.	Result
1	Check all the field by adding Correct data in the field	Filled All the Fields	Add Assessment to the Assessment module	Pass
2	Check the field by leaving the empty field and without question	Assessment type: Not Selected Question: Empty (If Question Added Leaving the Empty Field of) Course Outcome: Not Selected	Form cannot be submitted. Submit button cannot be clicked	Pass

Unit Testing 8: Import Data by using Appropriate Fields

Testing Objective: To ensure the data import is done Correctly is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Clicking the Download template	N/A	The template will be downloaded in excel format	Pass
2	Clicking the import button to import the filled template which is downloaded	N/A	Excel File is imported Correctly by clicking on view button all the data will be displayed on pop-up model	Pass

Unit Testing 9: ESE Data import by using Appropriate Fields

Testing Objective: To ensure the ESE data import is done Correctly is working Correctly.

No.	Test case/Test script	Attribute and value	Expected result.	Result
1	Clicking the Download template	N/A	The template will be downloaded in excel format	Pass
2	Clicking the import button to import the filled template which is downloaded	N/A	Excel File is imported Correctly by clicking on view button all the data will be displayed on pop-up model	Pass

5.2 Integration Testing

This section assesses whether a set of classes that must work together do so without error. They ensure that the interfaces and linkages between different parts of the system work properly.

Integration Testing 1: Add Faculty in system.

Testing Objective: To ensure the faculty is being added correctly and the **Interface** *is running correctly*.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Add Faculty	User Add faculty by providing all the detail of faculty	Successfully add Faculty to the faculty module and also saved it into real-time database.	Faculty added Successfully	Pass
2.	Update the Faculty	Edit the form by clicking update button and update the desired field which wants to be updated	The faculty has updated successfully and also updated the real-time database.	Faculty Updated successfully	Pass
3.	Delete the Faculty	Delete the faculty it will remove faculty from the module	The faculty remove from module and from database	Faculty Deleted Successfully	Pass

Integration Testing 2: Add Curriculum in system.

Testing Objective: To ensure the Curriculum is being added correctly and the **Interface** *is running correctly*.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Add Batch	User Add Batch by providing all the detail of Batch	Successfully added the batch into module and saved it into real-time database.	Batch added Successfully	Pass
2.	Add Term	User Add Term by providing all the detail of Batch	Successfully added the Term into module and saved it into real-time database.	Term added Successfully	Pass
3.	Add Course	User Add Course by providing all the detail of Batch	Successfully added the Term into module and saved it into real-time database.	Course added Successfully	Pass
4.	Add Course outcome	User Add Course outcome by providing all the detail of Batch	Course outcome added successfully and save to real-time database.	CO Added Successfully	Pass
5.	Mapping of CO/PO	Edit the CO/PO Mapping change the mapping	Its updated or mapped successfully	Mapped Saved	Pass

Integration Testing 3: Add Assessment in system.

Testing Objective: To ensure the Assessment is being added correctly and the **Interface** *is running correctly*.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Course Assessment	Select Curriculum, Term, and appropriate Course	Curriculum, Term, and Course is showing Correctly	Curriculum, Term and Course data is Selectable from the Options	Pass
2.	Add Assessment	User will add assessment to relevant course, assign type of assessment and add question to assessment, and mapping to its relevant CO	Assessment will be created and added to assessment module and in database	Assessment added Successfully	Pass

Integration Testing 4: Check Attainment in system.

Testing Objective: To ensure the Attainment is being checked correctly and the **Interface** *is running correctly*.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Data Import	Select Curriculum, Term, and appropriate Course and Download Template and then Fill and import the downloaded template	In view button the import will be showing	Imported data showed correctly in Pop-up model	Pass
2.	Exam Data Import	Select Curriculum, Term, and appropriate Course and Download the ESE Template and then Fill and import the downloaded template	ESE marks import successfully and will be viewed	Imported ESE marks showed correctly in Pop-up model	Pass
3.	CO attainments	Select Curriculum, Term, and appropriate Course and set the benchmark the calculated the attainments by clicking calculate button	Curriculum, Term, and Course is Selected Correctly and benchmark is also added it will show the calculated result of CIA attainments and ESE attainments	CIA and ESE attainment calculated and shown successfully	Pass

6 Conclusion and Future Work

This chapter concludes the project and highlights future work.

6.1 Conclusion

The Outcome Based Education web application aims to provide a comprehensive and seamless solution for education institutions to manage their curriculums, assessments, and outcomes. The application has been designed keeping in mind the requirements of both students and educators. The architecture of the application, including the 3-tier design, has been carefully planned to ensure that it is scalable, secure, and reliable. The application makes use of modern technologies like MERN Stack, MongoDB, and has been designed with a focus on user experience. The application is equipped with various features such as the ability to add and manage assessments, view student performance, and track the progress of student outcomes.

The application has undergone extensive testing to ensure that it meets all the necessary requirements. The use of unit tests has ensured that each component of the application functions correctly and provides expected results. The application's data management has been designed using MongoDB, providing a scalable and efficient solution for data storage and retrieval.

In conclusion, the Outcome Based Education web application is a comprehensive solution that provides a modern and user-friendly interface for both students and educators. The application has been designed to meet the needs of education institutions, and its architecture has been carefully planned to ensure that it is scalable, secure, and reliable.

6.2 Future Work

Future work for the Outcome Based Education web application may involve adding new features or functionalities to improve the user experience and meet evolving educational needs. Some potential areas of focus could include:

- Integration with other educational tools and platforms
- Gamification elements to make the learning process more engaging.
- Improved assessment and evaluation systems

- Personalized learning paths and recommendations
- Automated grading and feedback systems
- Mobile compatibility to allow for flexible access to educational resources.
- Expansion of the content library to include a wider range of subjects and disciplines.
- Improved data analytics to track student progress and make data-driven decisions.
- Enhanced collaboration and communication tools for students and teachers.

In order to ensure the continued success of the application, it will be important to regularly gather feedback from users and implement improvements based on their needs and suggestions.

8. References

Article in a Journal

De Guzman, M. F. D., Edaño, D. C., & Umayan, Z. D. (2017). Understanding the Essence of the Outcomes-Based Education (OBE) and Knowledge of its Implementation in a Technological University in the Philippines. *Asia Pacific Journal of Multidisciplinary Research, 5(4).*

World Wide Web

Akir, O., Eng, T. H., & Malie, S. (2012). Teaching and learning enhancement through outcome-based education structure and technology e-learning support. Procedia-Social and Behavioral Sciences, https://www.sciencedirect.com/science/article/pii/S1877042812034568?via%3Dihub

Harden, R. M. (2007). Outcome-based education: the future is today. Medical Teacher, 29(7), 625-629 https://www.tandfonline.com/doi/full/10.1080/01421590701729930