

A  
PROJECT REPORT  
ON  
**A TOOL TO DEVELOP OUTCOME BASED  
EDUCATION SYSTEM**

SUBMITTED TO THE SHIVAJI UNIVERSITY, KOLHAPUR  
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT  
FOR THE AWARD OF DEGREE  
OF

**BACHELOR OF TECHNOLOGY**  
in  
**Computer Science and Engineering**

SUBMITTED BY

<b>Ms. Snehal Ashok Mankar</b>	<b>29</b>	<b>21UGCS20708</b>
<b>Ms. Ananya Avadhut Rane</b>	<b>52</b>	<b>22UDCS21301</b>
<b>Ms. Asmita Vishvas Jadhav</b>	<b>69</b>	<b>21UGCS20830</b>
<b>Ms. Shruti Tanaji Nevarekar</b>	<b>74</b>	<b>21UGCS20796</b>

**PROJECT GROUP NO : A-01**

UNDER THE GUIDANCE OF

**Dr. R. B. Patil**



Department of Computer Science and Engineering  
SWVSM's

**Tatyasaheb Kore Institute of Engineering and  
Technology (Autonomous), Warananagar**  
Academic Year 2024-25



SWVSM's

**Tatyasaheb Kore Institute of Engineering and  
Technology (Autonomous), Warananagar**

## **CERTIFICATE**

This is to certify that the Project Report entitled,

**“A Tool To Develop Outcome Based Education System”**

Submitted by

<b>Name</b>	<b>Exam Seat No.</b>
<b>Ms. Snehal Ashok Mankar</b>	<b>21UGCS20708</b>
<b>Ms. Ananya Avadhut Rane</b>	<b>22UDCS21301</b>
<b>Ms. Asmita Vishvas Jadhav</b>	<b>21UGCS20830</b>
<b>Ms. Shruti Tanaji Nevarekar</b>	<b>21UGCS20796</b>

is a bonafide work carried out and is approved for the partial fulfillment of the requirement of Shivaji University, Kolhapur for the award of Degree of Bachelor of Technology in Computer Science and Engineering. This project work is a record of students' own work, carried out by them under our supervision and guidance during academic year 2024-25.

**Project Guide**

**HoD, CSE**

**Principal**

**Date :-**

**Place :- Warananagar**

**External Examiner**

## **ACKNOWLEDGEMENT**

This space is dedicated to acknowledge all those who have helped in bringing this project to fruition.

We are greatly indebted to our guide **Prof. R. B. Patil**, for his unstinted support and valuable suggestions. We are grateful to him not only for the guidance, but also for his unending patience and keeping our spirits high throughout. We express our sincere thanks to our beloved Head of the Department, **Prof. R. B. Patil** and Principal, for being source of inspiration and providing us the opportunity to work on this project.

We extend heartfelt thanks to all the teaching and non-teaching staff of the department of Computer Science and Engineering for their assistance and cooperation.

Finally, we would like to thank our parents and friends for their moral support and encouragement throughout our academics.

**Student Name**

**Exam Seat No.**

**Sign**

**Ms. Snehal Ashok Mankar**

**Ms. Ananya Avadhut Rane**

**Ms. Asmita Vishwas Jadhav**

**Ms. Shruti Tanaji Nevarekar**

**Date :-**

**Place :-** Warananagar

## **ABSTRACT**

This project explores the implementation and impact of Outcome-Based Education (OBE) as a transformative framework in modern educational systems. OBE emphasizes the achievement of specific, measurable learning outcomes by aligning curriculum design, teaching methodologies, and assessment strategies with clearly defined educational objectives. The approach fosters continuous improvement, accountability, and transparency, ensuring that instructional practices evolve to meet the diverse needs of learners and the dynamic demands of industry and society. Through its focus on equity and inclusion, OBE provides personalized learning experiences that empower all students to achieve success, regardless of their backgrounds or learning styles. This project examines how integrating OBE principles enhances student engagement, ensures academic and professional readiness, and equips graduates with the essential skills and competencies needed to thrive in a competitive global environment. Ultimately, the adoption of OBE contributes to the development of a more effective, efficient, and inclusive education system, reinforcing its role as a cornerstone of innovation and excellence in teaching and learning.

## **TABLE OF CONTENTS**

<b>CHAPTER</b>	<b>NAME</b>	<b>PAGE</b>
	<b>Certificate</b>	
	<b>Acknowledgement</b>	
	<b>Abstract</b>	<b>I</b>
	<b>List of Figures</b>	<b>II</b>
	<b>List of Tables</b>	<b>III</b>
	<b>Abbreviation</b>	<b>IV</b>
<b>1</b>	<b>Introduction</b>	<b>1 - 4</b>
	1.1 Introduction	1
	1.2 Motivation	3
<b>2</b>	<b>Literature Survey</b>	<b>5 - 8</b>
	2.1 Existing System	5
	2.2 Proposed System with Block Diagram	7
	2.3 Feasibility Study	8
<b>3</b>	<b>Project Scope and Requirement Analysis</b>	<b>9 - 10</b>
	3.1 Project Scope	9
	3.2 Requirement Gathering and Analysis	10
<b>4</b>	<b>Project Design and Modeling Details</b>	<b>11 - 29</b>
	4.1 Software Requirement Specification (SRS)	11
	4.2 System Modules	14
	4.3 System Modeling & Design	17
	4.4 Database Design	26
	4.5 System Architecture	28

<b>5</b>	<b>Implementation and Coding</b>	<b>30 - 38</b>
	5.1 Algorithms	30
	5.2 Flowcharts	33
<b>6.</b>	<b>Testing</b>	<b>39 - 43</b>
	6.1 Fundamentals of Testing	39
	6.2 Test Plan of the Project	41
	6.3 Test Cases and Test Result	43
<b>7.</b>	<b>Project Plan &amp; Schedule</b>	<b>54 - 55</b>
	7.1 Gantt Chart	54
<b>8.</b>	<b>Risk Management and Analysis</b>	<b>56 - 57</b>
	8.1 Project Risk Analysis	56
<b>9.</b>	<b>Configuration Management</b>	<b>58 - 66</b>
	9.1 Installatio/ Uninstallation	58
	9.2 User manual	61
<b>10</b>	<b>Conclusion and Future Scope</b>	<b>67- 66</b>
	10.1 Conclusion	64
	10.2 Future Scope	65

## **References**

- a) Journal/Conference Paper
- b) Book References
- c) Web Reference

## **LIST OF FIGURES**

<b>Figure No.</b>	<b>Name</b>	<b>Page</b>
Fig. 2.1	Block Diagram	7
Fig 4.1	Object-Class Diagram	17
Fig 4.2	Use case Diagram	19
Fig 4.3	Sequence Diagram	21
Fig 4.4	Activity Diagram	23
Fig 4.5	E-R Diagram	26
Fig 4.6	System Architecture	28
Fig 5.1	Flowchart	33
Fig 7.1	Gantt Chart	54

## **ABBREVIATIONS**

<b>Abbreviation</b>	<b>Name</b>
OBE	Outcome-Based Education
NBA	National Board of Accreditation
NAAC	National Assessment and Accreditation Council
PEO	Program Educational Objectives
PO	Program Outcomes
PSO	Program Specific Outcomes
CO	Course Outcomes
BoS	Board of Studies
GAs	Graduate Attributes
FV	Frequency-Variability
CIE	Continuous Internal Evaluation
ESE	End Semester Examination
ISE	Internal Assessment
TW	Term Work
NA	Not Applicable
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization



# Chapter 1

## Introduction

---

### 1.1 Introduction

In today's rapidly evolving educational landscape, Outcome-Based Education (OBE) stands out as a transformative approach designed to achieve specific, measurable learning outcomes. By emphasizing the alignment of curriculum and assessment with clearly defined objectives, OBE enhances clarity, accountability, and transparency within the educational process. This framework fosters a culture of continuous improvement, ensuring that teaching methodologies and assessment strategies are consistently refined to meet learners' diverse needs and evolving industry standards. Furthermore, OBE promotes equity and inclusion by tailoring learning experiences to accommodate individual differences, enabling all students to succeed. By integrating the principles of OBE, educational institutions aim to deliver engaging, impactful, and effective learning experiences that not only prepare students for future academic and professional success but also empower them to contribute meaningfully to society. This holistic approach ensures that graduates are equipped with the knowledge, skills, and competencies needed to thrive in an increasingly dynamic and competitive world. As a result, OBE has become a cornerstone of modern education, driving innovation and excellence in teaching and learning. By adopting OBE, educational institutions can ensure that their students are well-prepared to meet the challenges of the 21st century and make a positive impact in their chosen fields. Ultimately, OBE is a powerful tool for shaping the best and brightest, providing them with the skills, knowledge, and competencies required to succeed in an ever-changing world. By focusing on outcomes, OBE helps to create a more effective, efficient, and equitable education system that benefits students, educators, and society as a whole.

### **1.1.1 Overview**

Outcome-Based Education (OBE) is a student-centered approach that focuses on achieving specific, measurable learning outcomes by aligning curriculum, instruction, and assessment with clearly defined objectives. It is widely used in engineering education, with support from accreditation bodies like ABET and NBA, which emphasize the development of industry-relevant skills and graduate employability.

OBE enhances accountability and promotes continuous improvement by regularly assessing student performance against Course Outcomes (COs) and Program Outcomes (POs). This data-driven process helps institutions refine teaching methods and curricula to ensure better learning results. However, challenges such as defining appropriate outcomes, aligning teaching strategies, designing assessments, and managing faculty workload remain significant. These issues are often intensified by limited training and concerns about academic freedom.

To address these challenges, many institutions are adopting online expert systems that enable efficient, role-based management of COs and POs. These systems streamline data collection, facilitate faculty feedback, and support informed decision-making, making OBE implementation more effective and less burdensome.

Ultimately, OBE aims to foster equity, inclusivity, and student success, while driving institutional quality and contributing to broader societal and professional progress.

## **1.2 Motivation**

The motivation for adopting Outcome-Based Education (OBE) arises from a vision to create a transformative educational environment that significantly benefits both learners and society. OBE shifts the focus of education from traditional content delivery to the achievement of clearly defined and measurable learning outcomes. This transformation supports the broader mission of educational institutions by ensuring that their core values—such as integrity, innovation, and social responsibility—are embedded in every aspect of teaching and learning. By centering education around specific learning objectives, OBE provides clarity and direction for students, allowing them to understand what is expected of them and how their progress will be evaluated.

Through this outcome-focused approach, OBE enhances student achievement by promoting deep learning, critical thinking, and the development of practical skills relevant to real-world challenges. It also encourages institutions to engage in continuous improvement, using assessment data to refine curricula, teaching methods, and student support services. Moreover, OBE plays a crucial role in fostering equity and inclusion by recognizing diverse learning needs and ensuring that all students have the opportunity to succeed regardless of their background. This leads to a more inclusive and supportive learning environment.

Ultimately, OBE aligns educational efforts with the long-term aspirations of institutions to contribute meaningfully to societal development. By producing graduates who are not only academically proficient but also socially and professionally competent, institutions fulfill their responsibility to advance educational excellence and drive positive change in the broader community. The adoption of OBE, therefore, represents a commitment to quality, accountability, and relevance in higher education.

### **1.2.1 Purpose**

The purpose of Outcome-Based Education (OBE) is to ensure that educational systems are designed to achieve specific, measurable learning outcomes that contribute to students' academic, professional, and personal growth. Unlike traditional education models, OBE shifts the focus from content delivery to a student-centered approach, where the primary goal is for students to demonstrate competence in specific skills and knowledge. This alignment of curriculum, instruction, and assessment with clearly defined outcomes ensures that students understand what is expected of them and how they will be evaluated. OBE enhances accountability by making both students and educators responsible for achieving these outcomes, which fosters transparency and efficiency in the educational process.

Moreover, OBE emphasizes the development of essential life skills such as critical thinking, communication, and problem-solving, preparing students not only for academic success but also for the workforce. It promotes continuous improvement within institutions by assessing student performance, which leads to the refinement of teaching methods and curricula to meet evolving industry standards. OBE also supports equity and inclusivity by accommodating diverse learning styles and paces, ensuring all students have equal opportunities to succeed. Furthermore, it aligns educational outcomes with societal and industry needs, making graduates more employable and better equipped to contribute to social and economic progress. Ultimately, OBE is about creating a system where students take ownership of their learning, are motivated by clear goals, and are prepared to meet the challenges of the modern world.

## **Chapter 2**

### **Literature Survey**

---

#### **2.1 Existing System**

Educators often rely on traditional tools like MS Office for curriculum development, lesson planning, and assessment, which can create significant inefficiencies in the teaching and learning process. While these tools may serve their purpose, they often lack the ability to provide a comprehensive, structured approach to align learning materials and assessments with specific learning outcomes, a key principle of Outcome-Based Education (OBE). This reliance on basic software can result in inconsistent alignment between what is taught and what is assessed, making it difficult to track students' progress toward achieving the desired learning goals. Additionally, traditional tools may not easily accommodate diverse learning needs, such as varying student abilities or different learning styles, nor do they provide the flexibility required to adapt to evolving standards in education. As a result, the use of these conventional methods can hinder the creation of a comprehensive and effective educational environment that OBE envisions—one that ensures all students meet clearly defined outcomes while continuously improving teaching strategies.

##### **2.1.1 Referred Journal/Conference Papers**

- I. IEEE Paper: <https://files.eric.ed.gov/fulltext/ED380910.pdf>
- II. IEEE Paper: [https://saudijournals.com/media/articles/JAEP\\_52\\_58-65.pdf](https://saudijournals.com/media/articles/JAEP_52_58-65.pdf)
- III. A HANDBOOK ON OUTCOME BASED EDUCATION (OBE) By: Team OBE Mr.  
(Ramgouda B. Patil, Mr. Mahesh A. Patil, Ms. S. V. Nikam ) Department of Computer  
Science & Engineering (TKIET)

### **2.1.2 Elaborate on Existing System Applications / Examples**

**MS Office:** In the existing educational system, administrators and educators often rely on traditional methods and tools like MS Office for curriculum development and assessment. These methods can be cumbersome and inconsistent, lacking alignment with specific learning outcomes. Manual processes for tracking and assessing student progress using such tools can lead to inefficiencies and a lack of clarity in measuring educational effectiveness. In the current educational system, administrators and educators often depend on traditional tools like MS Office for curriculum development and assessment, which can be labour-intensive, inconsistent, and poorly aligned with specific learning outcomes. These manual processes make tracking and evaluating student progress inefficient and hinder the ability to measure educational effectiveness clearly. Additionally, such methods struggle to accommodate diverse learning needs and keep pace with evolving educational standards, posing challenges in fostering a comprehensive and equitable learning environment as envisioned by Outcome Based Education (OBE).

### **2.1.3 Limitations or Challenges in Existing Systems**

1. **Lack of Alignment:** Manual methods and MS Office tools may not align well with specific learning outcomes.
2. **Inefficiency:** The process of creating and managing educational content manually is time consuming and error- prone.
3. **Difficulty in Tracking Progress:** Manual tracking and assessment hinder effective monitoring of student development.
4. **Limited Adaptability:** Existing methods struggle to address diverse learning needs and evolving standards.
5. **Collaboration Challenges:** Managing multiple contributors can lead to inconsistencies and confusion.

## 2.2 Proposed System with block diagram

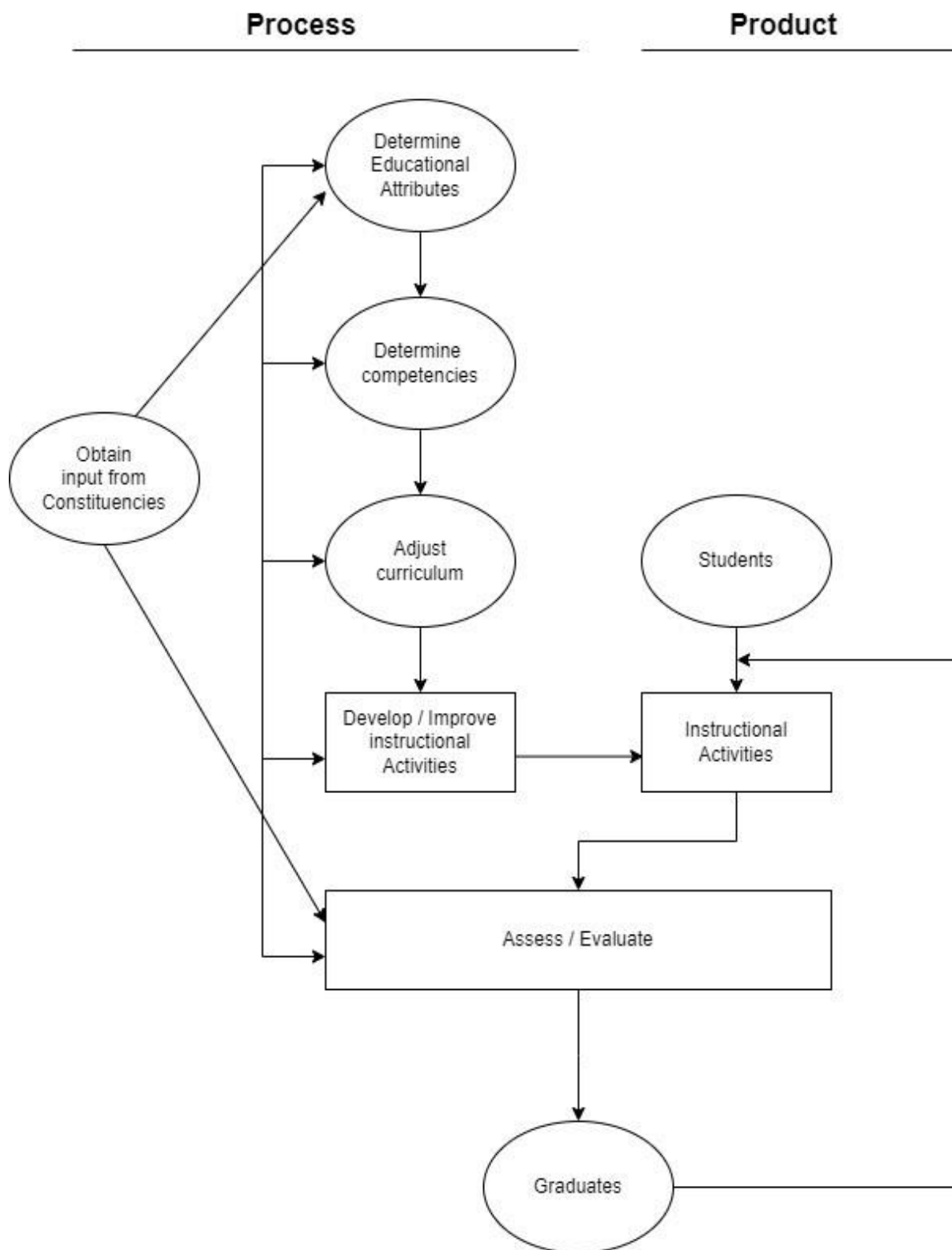


Fig 2.1 Block diagram

## 2.3 Feasibility Study

### 1. Technical Feasibility

- **Technology Stack:** Utilizes Java, Spring Boot, MySQL, and Apache Tomcat. The technology stack is suitable for developing a scalable and robust web application.
- **System Requirements:** Minimum hardware includes an Intel Core i5 processor, 8 GB RAM, and 256 GB SSD. Recommended hardware includes higher specifications for better performance.
- **Deployment Environment:** Requires a web server (Apache Tomcat), and optional use of Docker for containerization. Ensures scalability and security.

### 2. Financial Feasibility

- **Development Costs:** Includes hardware, software licenses (if needed), and development team salaries. Open-source tools reduce initial costs.
- **Operational Costs:** Includes server maintenance, database management, and software updates. Use of cloud-based infrastructure can optimize costs.
- **Return on Investment (ROI):** Improved efficiency in curriculum management and student assessment is expected to lead to cost savings and enhanced educational outcomes.

### 3. Operational Feasibility

- **Development and Maintenance:** Requires expertise in Java, Spring Boot, and web development. The development team should be capable of managing the project through its lifecycle.
- **User Adoption:** Potential challenges include resistance to change and the need for comprehensive training. Effective training and support will be necessary for successful implementation.
- **Scalability:** The system is designed to scale with increasing user load and evolving educational standards.

### 4. Legal and Regulatory Feasibility

- **Compliance:** The tool must comply with data protection regulations such as GDPR.
- Regular updates and compliance checks will be necessary.
- **Licensing:** Use of open-source technologies helps mitigate licensing costs, but careful review of third-party software licenses is required.



## Chapter 3

# Project Scope and Requirement Analysis

---

### 3.1 Project Scope

The project focuses on implementing an Outcome-Based Education (OBE) management system. The system streamlines institutional processes by enabling administrators, faculty, and HODs to align academic activities with defined outcomes. It includes modules for creating programs, assigning credits, defining outcomes, mapping assessments, and monitoring attainment to improve transparency, accountability, and academic performance.

#### In-Scope

- **Principal's Module:** Define vision, mission, create programs, and assign HODs.
- **Dean's Module:** Assign program/semester credits and format question papers.
- **HOD's Module:** Define vision, mission, outcomes (POs, PEOs, PSOs), map COs, and assign courses.
- **Faculty's Module:** Upload curriculum, marks, and calculate attainment.
- **Admin Module:** Monitor system operations.

#### Out-of-Scope

- Integration with external systems (e.g., LMS or ERP).
- Non-academic modules like HR or finance.
- Real-time analytics and accreditation automation.

#### Boundaries

- **Users:** Admin, principal, dean, HODs, and faculty.
- **Focus:** OBE processes and manual data input for curriculum, assessments, and mapping.

#### Deliverables

- Functional modules for all roles.
- Tools for outcome mapping and attainment calculations.
- Central monitoring and attainment reports.
- User manuals and system documentation.

## 3.2 Requirement Gathering & Analysis

### 3.2.1 Development Environment :-

#### (1) Software Requirements:

- OS: Windows (64-bit) (10+)
- Back End Server: MySQL
- Programming Language: PHP

#### (2) Hardware Requirements

- Processor: Core i3 and above
- RAM: 8GB and above
- Storage: 100GB and above

### Deployment Environment:-

#### (1) Software Requirements:

- OS: Windows (64-bit)(10+)
- Web Server: Apache Tomcat, MySQL Server
- Database: MySQL

#### (2) Hardware Requirements:

##### Minimum:

- Processor: Core i3 and above
- RAM: 8 GB.
- Storage: 256GB SSD.

##### Recommended:

- Processor: Intel Core i7/i9 or AMD Ryzen.
- RAM: 16 GB.
- Storage: 512GB SSD.

## Chapter 4

# Project Design and Modeling Details

---

### 4.1 Software Requirement Specification (SRS)

#### 1. Introduction

- Purpose: Define the requirements for a web application to support Outcome Based Education (OBE) by aligning curriculum, instruction, and assessment with measurable outcomes.
- Scope: The tool will facilitate curriculum development, course management, and student assessment in an OBE framework.
- Definitions: Terms like Outcome-Based Education (OBE), curriculum, assessment, etc., are defined for clarity.

#### 2. Overall Description

- Product Perspective: The application will be a web-based solution with a userfriendly interface for faculty and administrative staff.
- Product Functions: Key functions include login/signup, department management, course management, assignment of courses, component management, and reporting.
- User Characteristics: Users include faculty, department heads, and administrators with varying levels of technical expertise.
- Constraints: Constraints include adherence to data protection regulations and integration with existing educational systems if required.

### 3. Specific Requirements

#### Functional Requirements:

- Login/Signup: Secure authentication for users.
- Department Management: CRUD operations for departments.
- Course Management: Create, update, delete, and view courses.
- Offer Course: Manage which courses are offered in which semester.
- Assign Course: Assign courses to faculty.
- Components Management: Manage course components and outcomes.
- Report Generation: Generate attainment sheets and other reports.

#### Non-Functional Requirements:

- Performance: The application should support up to 500 concurrent users.
- Usability: The interface should be intuitive and accessible.
- Reliability: The system should have 99.9% uptime.
- Security: Data should be encrypted and protected from unauthorized access.

#### External Interface Requirements:

- User Interfaces: Web-based interfaces using HTML5, CSS3, and JavaScript.
- Hardware Interfaces: The application will run on standard web servers.
- Software Interfaces: Integration with MySQL for database management and Apache Tomcat for server hosting.
- Communication Interfaces: Use of HTTP/HTTPS for data transmission.

#### 4. System Features

- Login/Signup: Secure access control for different user roles.
- Department Management: Functionality to manage and organize academic departments.
- Course Management: Tools for creating and managing courses and their details.
- Offer Course: Scheduling and assignment of courses for academic terms.
- Assign Course: Allocation of courses to faculty members.
- Components Management: Managing course components and mapping to learning outcomes.
- Report Generation: Automated generation of reports for student attainment and progress.

#### 5. Design Constraints

- The system must comply with educational data protection laws.  
The solution must be compatible with existing IT infrastructure where applicable.

#### 6. Assumptions and Dependencies

- Assumes availability of a stable internet connection for users.
- Depends on the continued support and updates for the technologies used (Java, Spring Boot, MySQL).

## **4.2 System Modules**

### **Principal's Module**

- **Vision and Mission:**

Allows the principal to define and update the institution's vision and mission statements. These are guiding principles for academic and administrative operations and are essential for accreditation and strategic planning.

- **Create Program (UG/PG):**

Enables the creation of undergraduate and postgraduate academic programs. This includes specifying the program name, duration, eligibility, and other foundational attributes.

- **Assign HOD:**

Facilitates the assignment of Heads of Departments (HODs) to various programs. The system maintains records of department leadership and helps in access control based on roles.

### **Dean's Module**

- **Assign Credits:**

The dean allocates course credits for each semester across different departments. This includes core, elective, lab, and project courses, ensuring academic workload balance.

- **Question Paper Formatting:**

Provides standardized templates and guidelines for setting internal and external exam papers. Ensures consistency in structure, marking scheme, and learning outcome alignment.

## HOD's Module

- **Department Vision and Mission:**

Allows HODs to define or align the department's specific vision and mission with those of the institution. These statements reflect the academic goals and values of each department.

- **Manage Semesters and Courses:**

Lets HODs configure the semester structure (like number of weeks, exam periods) and manage courses offered each term, including elective pools and subject codes.

- **Upload Outcomes:**

Facilitates uploading and mapping of:

- **COs (Course Outcomes)**
- **POs (Program Outcomes)**
- **PEOs (Program Educational Objectives)**
- **PSOs (Program Specific Outcomes)**

This supports Outcome-Based Education (OBE) and accreditation processes like NBA/NAAC.

- **Assign Courses:**

HODs can assign courses to faculty members based on specialization, workload, and experience. Tracks who is responsible for which subject each term.

### **Faculty's Module**

- **Upload Curriculum:**

Faculty members can upload detailed theoretical and practical course content, including weekly plans, lab manuals, references, and learning objectives.

- **Upload Marks:**

Records student performance for various evaluations like assignments, quizzes, midterms, practicals, and finals. Supports mark entry, result calculation, and reporting.

- **Attainment:**

Calculates course attainment based on student performance, surveys, and feedback. Maps attainment to COs and POs, and supports continuous improvement strategies.

### **Admin User Module**

- **Monitor System:**

Admin users have system-wide access to monitor performance, resolve issues, manage users, generate logs, and ensure data integrity. They can also back up data, update configurations, and manage user roles and access permissions



## 4.3 System Modeling & Design

### 4.3.1 Object-Class Diagram

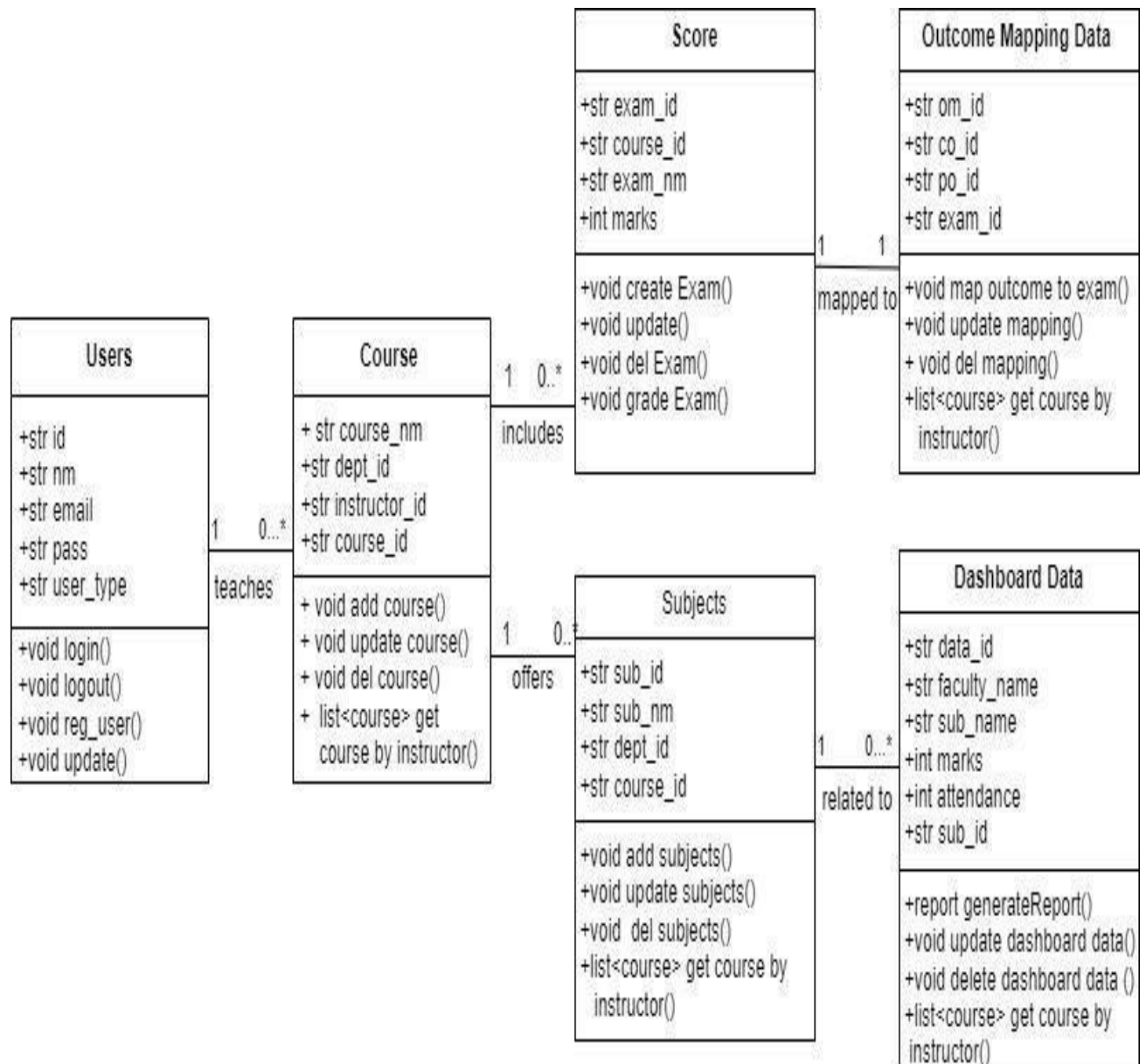


Fig 4.1 Object-Class Diagram

The class diagram for an Outcome-Based Education (OBE) system presents a structured model that integrates users, academic programs, subjects, assessments, outcome mappings, and performance monitoring to ensure effective education delivery. At the core is the users class, responsible for managing authentication, user roles, and profiles, distinguishing among principals, deans, HODs, faculty, and admin users. These users interact directly with the courses class, which represents undergraduate and postgraduate academic programs. Each course includes essential attributes like name, type, duration, and total credits, and is managed through functions for creating, updating, deleting, and retrieving course information.

Each course contains multiple subjects, represented by the subjects class. This class includes subject-specific details such as ID, name, credit value, semester, and course linkage. Subjects are closely tied to the scores class, which records student performance in various assessments, including internal and external exams. The scores class supports operations for recording, updating, and grading student marks, enabling a consistent evaluation process.

A vital part of the system is the outcome mapping data class. This class enables the core functionality of OBE by linking course outcomes (COs) and program outcomes (POs) to specific assessments. It uses mapping IDs, score IDs, and weightage to track how well students achieve targeted learning outcomes. The data from this mapping is essential for calculating attainment levels and ensuring that academic assessments align with institutional goals.

The dashboard data class serves as the monitoring layer of the system, aggregating information from the outcome mappings and scores. It tracks faculty performance and academic effectiveness by displaying metrics such as average marks, feedback scores, and CO/PO attainment levels. This class helps faculty and administrators review performance data and implement improvements. Overall, the class diagram supports the alignment of teaching, assessment, and learning outcomes, forming the backbone of a data-driven, outcome-oriented education system.

## 4.3.2 Use case diagram

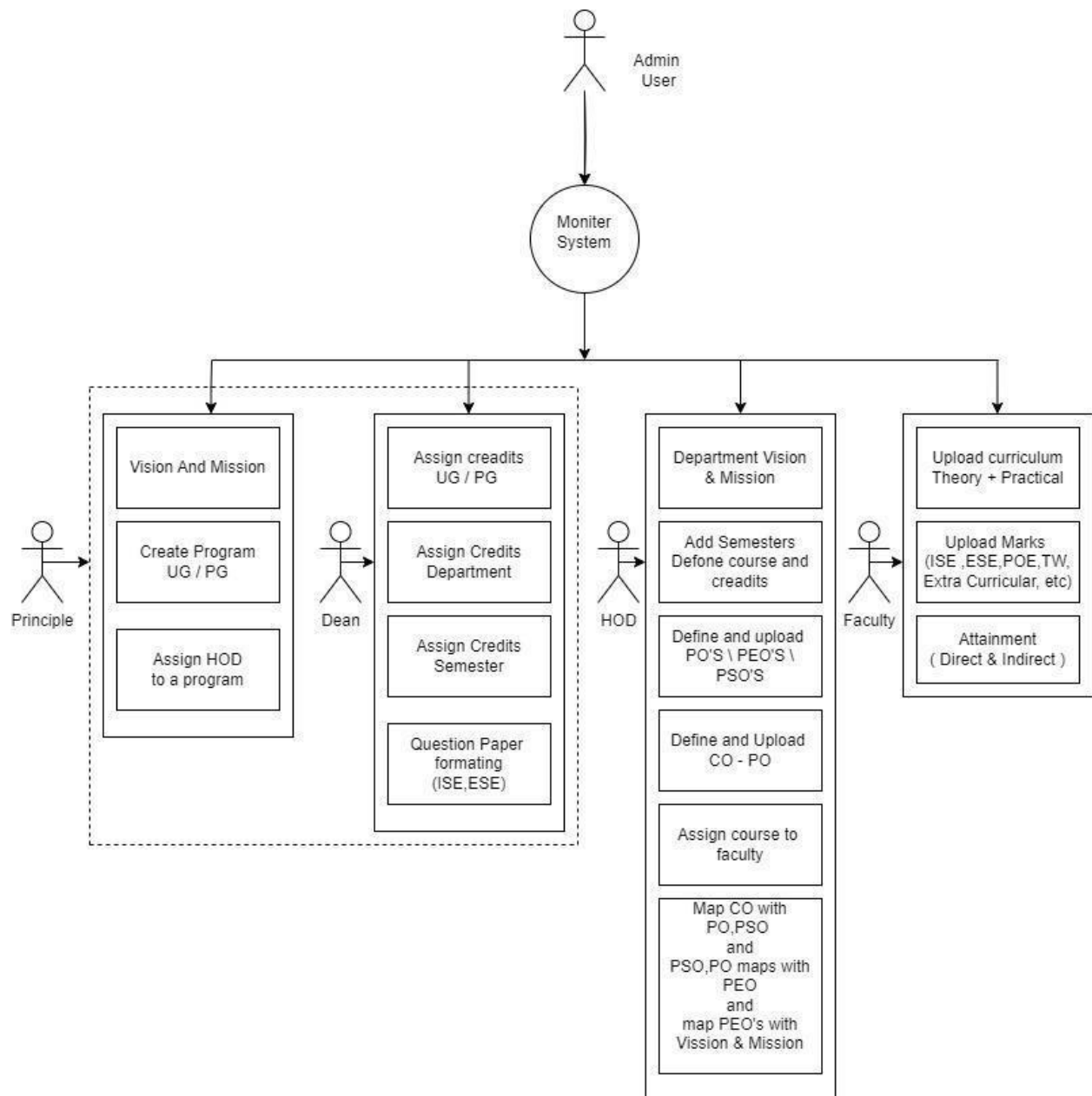


Fig 4.2 Use Case Diagram

This use case diagram outlines the workflow of an Outcome-Based Education (OBE) system, highlighting the roles of key stakeholders: the principal, dean, head of department (HOD), faculty, and admin user. Each role contributes to the academic and operational effectiveness of the system, ensuring alignment with institutional goals and educational outcomes.

The principal plays a strategic role by setting the institution's vision and mission, creating undergraduate and postgraduate programs, and assigning HODs to manage them. These foundational steps shape the academic structure and leadership of the institution. The dean supports program implementation by assigning course credits across departments and semesters and ensuring standardized exam formatting for consistency in assessments.

The HOD oversees departmental functions by defining the department's vision and mission, assigning courses to faculty, and managing academic outcomes. This includes uploading and maintaining program outcomes (POs), course outcomes (COs), program educational objectives (PEOs), and program-specific outcomes (PSOs), which are central to outcome-based assessment and accreditation.

Faculty members are responsible for the daily execution of teaching and evaluation. They upload curriculum content, record student marks, and track attainment by mapping student performance to COs, POs, PEOs, and PSOs. This ensures that assessments directly reflect learning objectives and support continuous improvement in teaching.

The admin user maintains the technical infrastructure of the system, managing user access, monitoring performance, and ensuring smooth operations across all modules. In summary, the use case diagram presents a collaborative and outcome-focused approach to academic management, integrating planning, instruction, assessment, and system support to achieve the goals of OBE.

### 4.3.3 Sequence Diagram

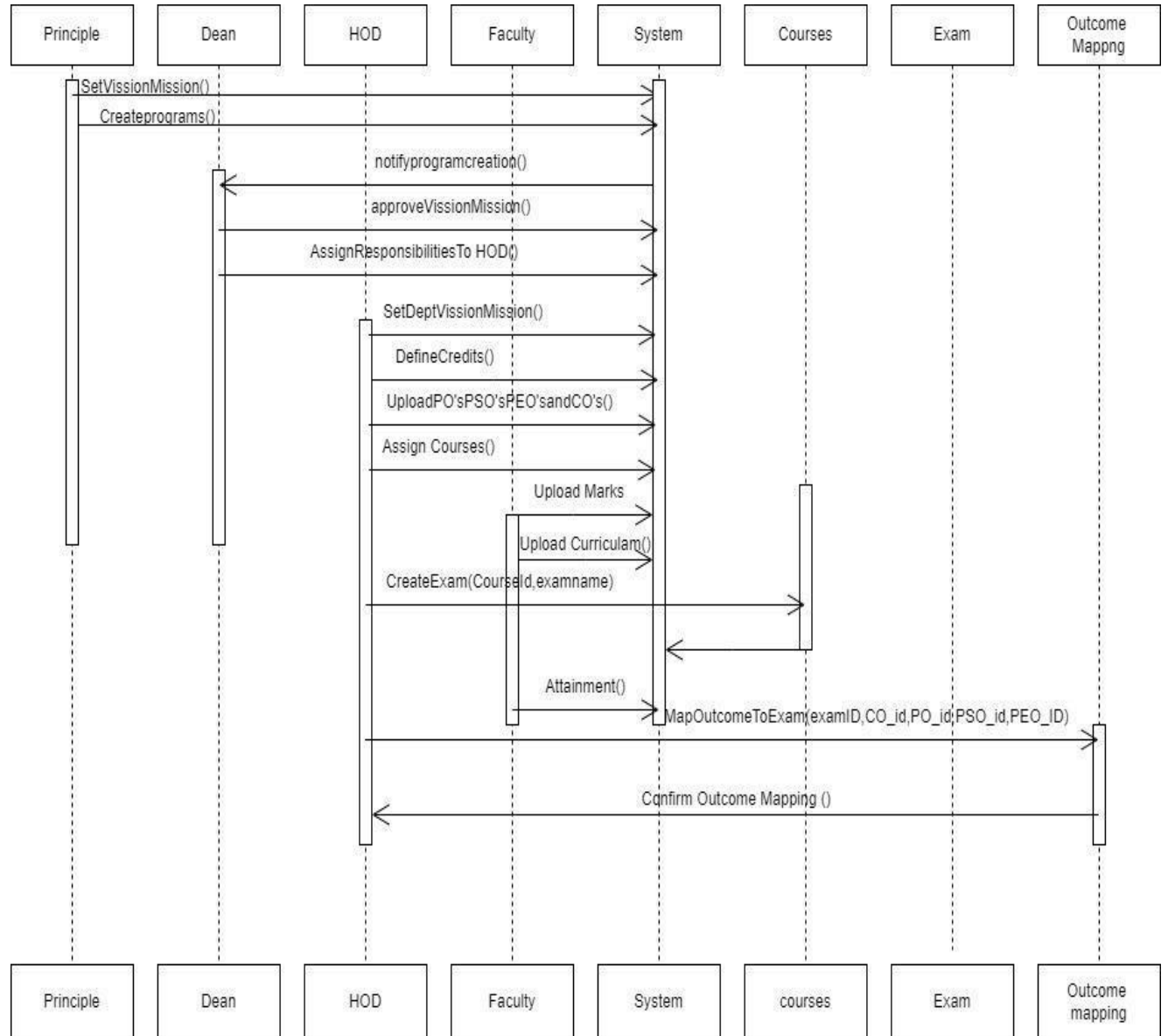


Fig 4.3 Sequence Diagram

The sequence diagram of the Outcome-Based Education (OBE) system illustrates the step-by-step interactions between key roles, including the principal, dean, head of department (HOD), faculty, and system components like Courses, Exam, and Outcome Mapping. The workflow begins with the principal setting the institution's vision and mission, which are shared with the dean for review and program creation. Based on this foundation, the dean sets up academic programs and delegates responsibilities to the HOD.

The HOD aligns the department's vision and mission with institutional goals and defines the academic structure by assigning course credits and uploading Program Outcomes (POs), Program Specific Outcomes (PSOs), Program Educational Objectives (PEOs), and Course Outcomes (COs) into the system. The HOD also assigns faculty members to specific courses, initiating the teaching and assessment phase.

Faculty members upload the curriculum, create exams aligned with the defined outcomes, and, after conducting assessments, enter student marks into the system. The system then maps these results to the corresponding COs, POs, PSOs, and PEOs, calculating attainment levels that reflect students' achievement of learning objectives. The outcome mapping is then reviewed and finalized, completing a cycle that ensures alignment between curriculum delivery, assessment, and educational goals.

This structured sequence reinforces the core principle of OBE—ensuring that all academic activities are planned and measured based on clearly defined outcomes, leading to continuous improvement and better educational quality.

### 4.3.4 Activity Diagram

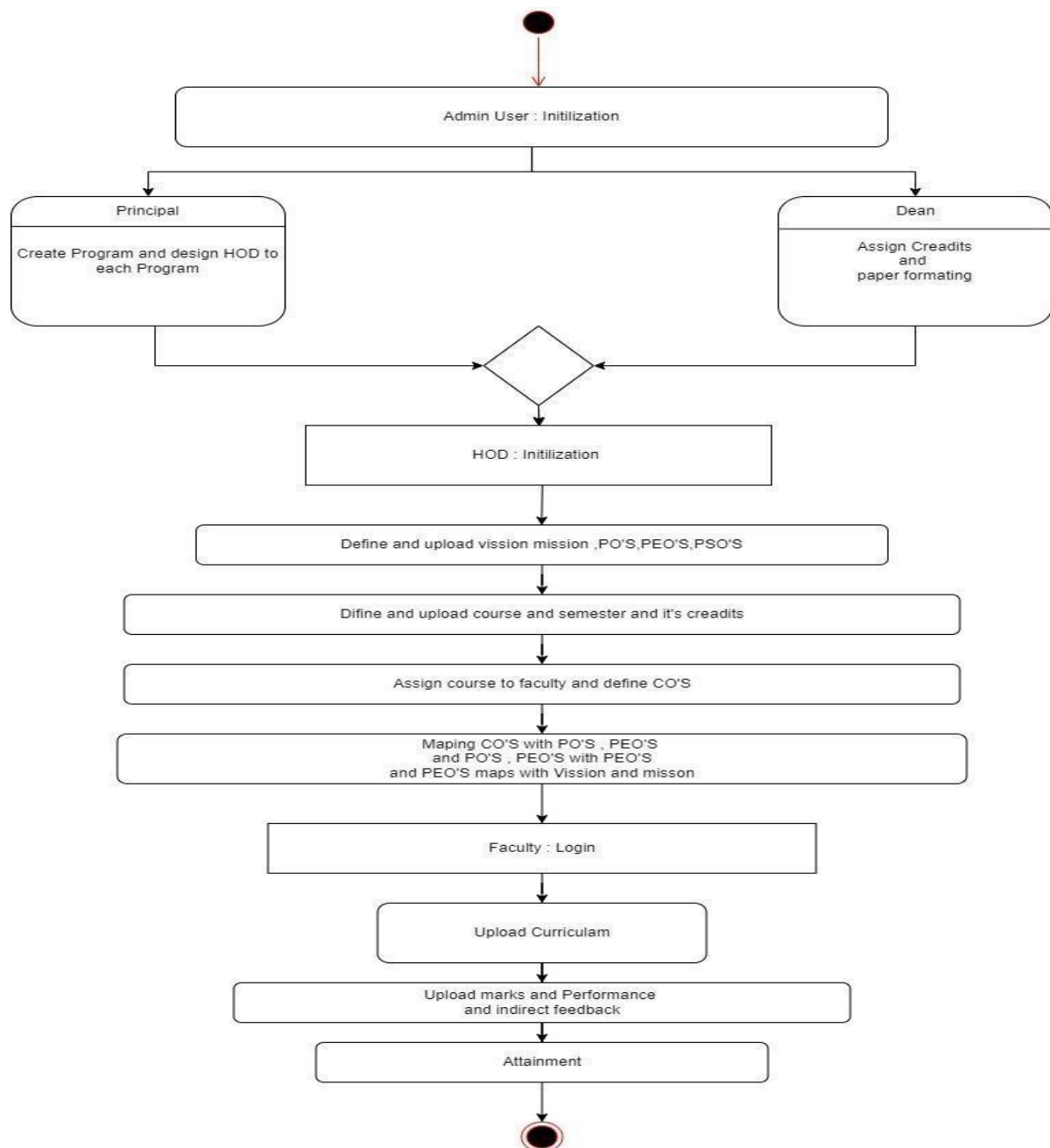


Fig 4.4 Activity Diagram

The Outcome-Based Education (OBE) system follows a structured workflow designed to ensure quality education and continuous improvement through a clear focus on outcomes. The process begins with the admin user who initializes the system by setting up the basic environment and creating user accounts for the principal, dean, heads of departments (HODs), and faculty members. The principal is responsible for creating academic programs, such as undergraduate or postgraduate degrees, and assigning HODs to manage these programs. Once the programs are established, the dean steps in to assign course credits in accordance with academic regulations and ensures that examination papers follow a standardized format to maintain consistency across courses.

Following this, the HOD plays a central role in shaping the academic and outcome structure of each program. The HOD defines and uploads the program's vision and mission statements, along with outlining the Program Outcomes (POs), Program Educational Objectives (PEOs), and Program Specific Outcomes (PSOs). These elements serve as the foundation for what the department aims to achieve through its educational offerings. The HOD also sets up the courses for each semester, assigns them to faculty members, and defines Course Outcomes (COs) that detail the specific knowledge and skills students should acquire by the end of each course. These COs are then systematically mapped to the broader POs, PEOs, and PSOs to ensure alignment between teaching activities and program goals.

Faculty members log into the system to access their assigned courses. They upload the course curriculum, teaching plans, and other instructional materials. Throughout the semester, they assess students using various tools such as quizzes, assignments, mid-semester tests, and final exams. They then upload the student performance data into the system, which is used to evaluate the attainment of the COs. Faculty also gather and submit indirect feedback from stakeholders like students, alumni, and employers, which provides qualitative insights into the effectiveness of the teaching and learning process.



The system uses this data to automatically calculate the attainment levels of course outcomes and their contributions to program outcomes. Quantitative methods are used to determine whether the predefined thresholds for COs, POs, PSOs, and PEOs have been met. Based on the analysis, areas where outcomes are weakly attained are identified. The department then formulates action plans to address these gaps, which might include curriculum revisions, updated teaching methods, or additional student support. This entire cycle supports continuous monitoring and improvement, a fundamental principle of the OBE system, ensuring that educational quality is consistently evaluated and enhanced.

## 4.4 Database Design

### ER Diagram

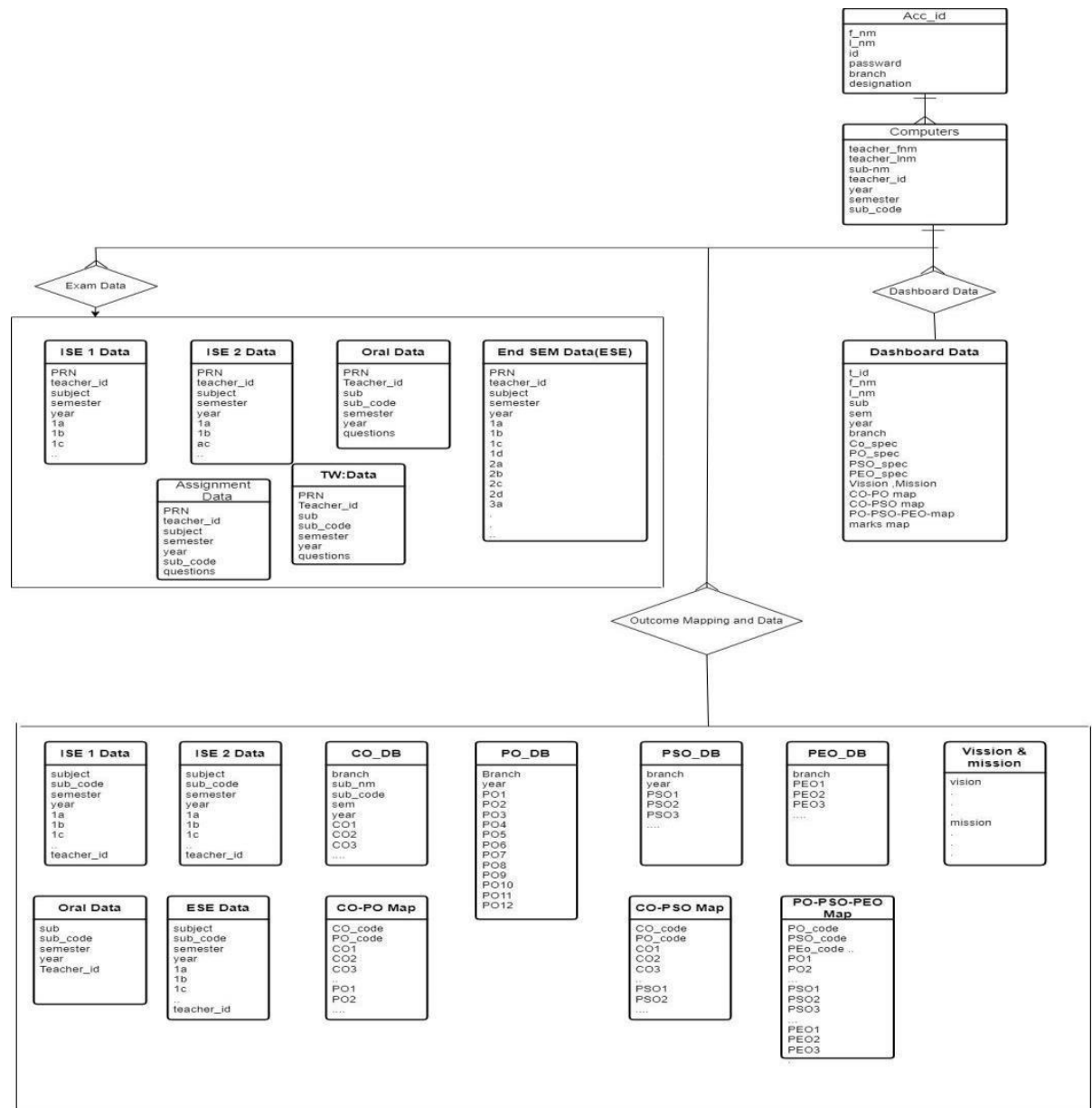


Fig 4.5 ER Diagram

The Entity-Relationship (ER) diagram for the Outcome-Based Education (OBE) system is structured to manage assessments, academic outcomes, and institutional objectives efficiently. A key component is the exam data entity, which stores detailed records of various assessments such as ISE 1, ISE 2, Oral exams, Term Work (TW), and End Semester Exams (ESE). Each record includes the student's PRN, teacher information, subject, semester, academic year, and specific question details. This allows the system to evaluate performance at both student and question levels.

The outcome mapping data connects Course Outcomes (COs) with Program Outcomes (POs), Program-Specific Outcomes (PSOs), and Program Educational Objectives (PEOs). This ensures that each assessment item contributes to a defined educational goal, enabling comprehensive outcome tracking. Teacher details are maintained through entities like `acc_id` and `computers`, which store faculty roles and subject assignments.

Dashboard data aggregates performance and outcome mappings to provide insights into attainment levels, helping institutions monitor progress and identify areas for improvement. The system also uses separate databases—`CO_DB`, `PO_DB`, `PSO_DB`, and `PEO_DB`—to organize and manage each type of outcome. This modular structure supports clarity, scalability, and continuous improvement within the OBE framework.

## 4.5 System Architecture

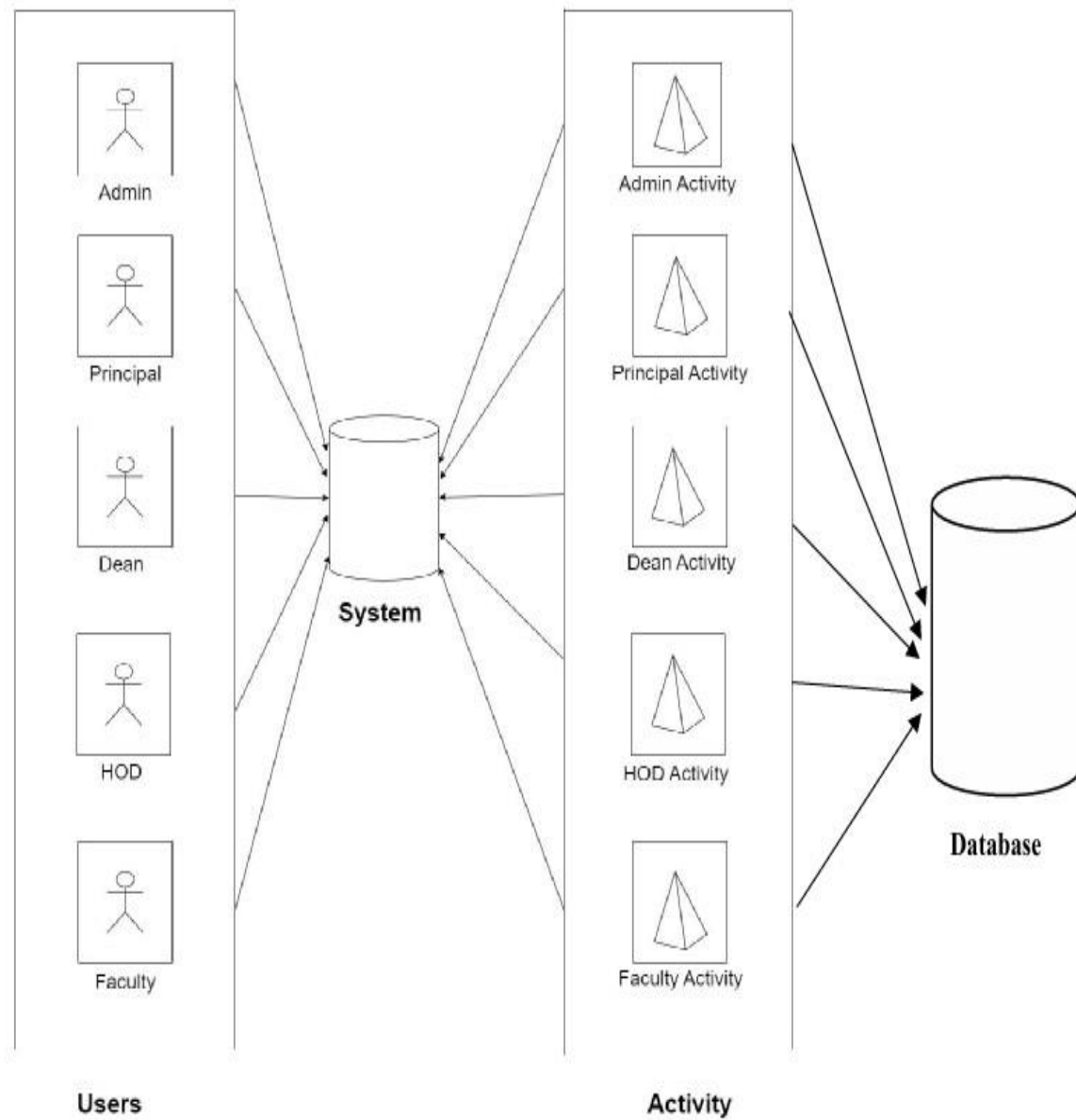


Fig 4.6 System Architecture

The system architecture of the Outcome-Based Education (OBE) system outlines how key academic and administrative roles—Principal, Dean, HOD, Faculty, and Admin—interact with a centralized database to manage institutional processes efficiently. The Principal initiates the academic framework by defining the institution's vision and mission and creating academic programs that align with long-term goals. These programs are passed to the Dean, who reviews the vision and mission and assigns specific responsibilities to the HODs. The Dean also oversees the structure and formatting of examinations like Internal Semester Exams (ISE) and End Semester Exams (ESE), ensuring consistency and academic integrity.

The Head of Department (HOD) plays a critical role at the department level. They establish a department-specific vision and mission, assign course credits, define Course Outcomes (COs) and Program Outcomes (POs), and allocate courses to faculty members. HODs also handle the mapping of COs to POs, ensuring each course contributes to the overall educational goals.

Faculty members are responsible for creating and conducting exams, uploading assessment marks, and grading students. Their input provides the performance data required to evaluate student progress and outcome attainment. Meanwhile, the Admin maintains the technical infrastructure, ensuring smooth operation of the system, managing user roles, and maintaining database integrity.

At the heart of the system is a centralized database that stores all key academic data, including courses, exams, outcomes, student marks, and attainment metrics. All user activities are integrated through this database, enabling accurate tracking, continuous monitoring, and informed decision-making. This interconnected system supports the core objectives of the OBE model—transparency, accountability, and continuous improvement in education.

## Chapter 5

### Implementation and Coding

---

#### 5.1 Algorithms

##### Step 1: Admin User Initialization

1. Start the system.
2. Admin logs into the system.
3. Initialize base configurations and settings.

##### Step 2: Principal Tasks

4. Principal logs in.
5. Create academic programs (UG/PG).
6. Assign a Head of Department (HOD) to each program.

##### Step 3: Dean Tasks

7. Dean logs in.
8. Assign credit structure for UG and PG programs.
9. Define formatting for internal (ISE) and end-semester exams (ESE).

##### Step 4: HOD Initialization

10. HOD logs in.
11. Verify assignments from the principal and dean.

##### Step 5: Department Setup by HOD

12. Define and upload:

- Department Vision and Mission
- Program Outcomes (POs)
- Program Educational Objectives (PEOs)
- Program-Specific Outcomes (PSOs)

13. Define and upload:

- Courses for the department
- Semester-wise structure
- Credit details for each course

#### **Step 6: Course and Outcome Mapping**

14. Assign courses to faculty.

15. Define Course Outcomes (COs) for each course.

16. Map:

- COs to POs and PSOs
- POs to PEOs
- PEOs to Vision and Mission

#### **Step 7: Faculty Tasks**

17. Faculty logs in.

18. Upload the curriculum content (theory and practical).

19. Upload:

- Student marks (ISE, ESE, practical, etc.)

- Student performance metrics
- Indirect feedback (surveys, exit interviews, etc.)

**Step 8: Attainment Calculation**

20. System calculates attainment:

- Direct attainment (based on marks/assessments)
- Indirect attainment (based on feedback)

21. Reports are generated for review.

**Step 9: End of Process**

22. Process ends and data is stored for continuous improvement and audit.



## 5.2 Flowchart

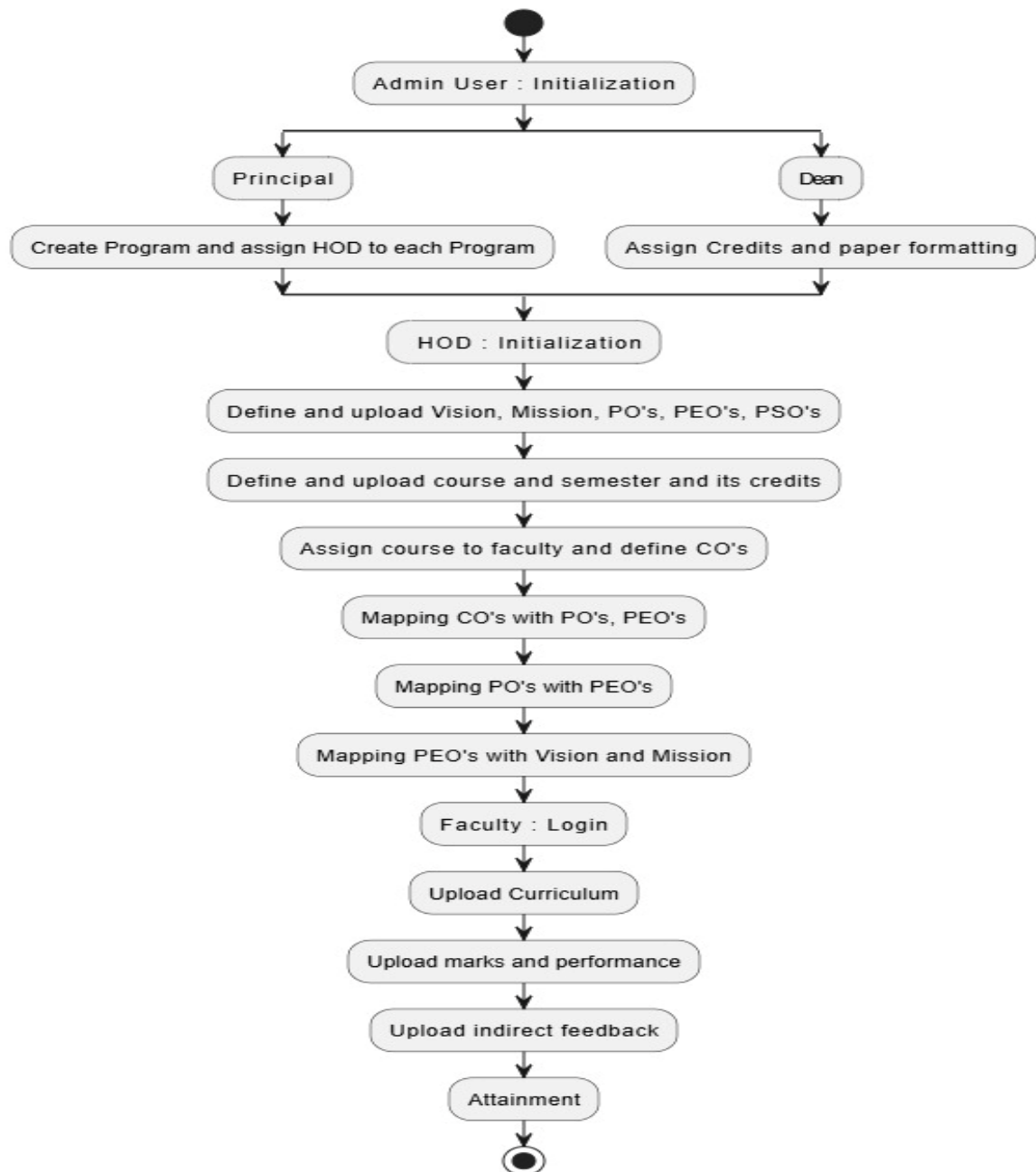


Fig 5.1 Flow Chart

### **1. Admin User: Initialization**

- The Admin starts the system and manages the platform setup.
- They create user accounts and grant appropriate access to the Principal, Dean, HOD, and Faculty.

### **2. Principal**

- Responsible for creating academic programs (UG/PG).
- Assigns HODs to respective programs, setting the academic leadership in motion.

### **3. Dean**

- Assigns academic credits to courses across departments and levels.
- Oversees question paper formatting (e.g., ISE, ESE formats) to maintain consistency.

### **4. HOD Initialization**

- Takes over after Principal and Dean complete their initial roles.
- Begins the academic framework definition for the department.

### **5. Define and Upload Vision, Mission, PO's, PEO's, PSO's**

- HOD uploads the departmental vision and mission.
- Defines and uploads:
  - Program Outcomes (POs)
  - Program Educational Objectives (PEOs)
  - Program Specific Outcomes (PSOs)

## **6. Define and Upload Course, Semester, and Credits**

- HOD sets up the semester-wise course structure along with the associated credit hours.

## **7. Assign Course to Faculty and Define CO's**

- Each course is assigned to a faculty member.
- Course Outcomes (COs) are defined for each course, aligning with broader program goals.

## **8. Outcome Mapping**

- COs mapped to POs and PEOs: Ensures course content aligns with expected program-level achievements.
- POs mapped to PEOs: Links learning outcomes to educational objectives.
- PEOs mapped to Vision and Mission: Ensures all educational activity supports institutional purpose.

## **9. Faculty Login**

- Faculty members access their assigned courses and responsibilities through the system.

## **10. Upload Curriculum**

- Faculty uploads theory and practical content for each course, making it accessible to students.

## **11. Upload Marks and Performance**

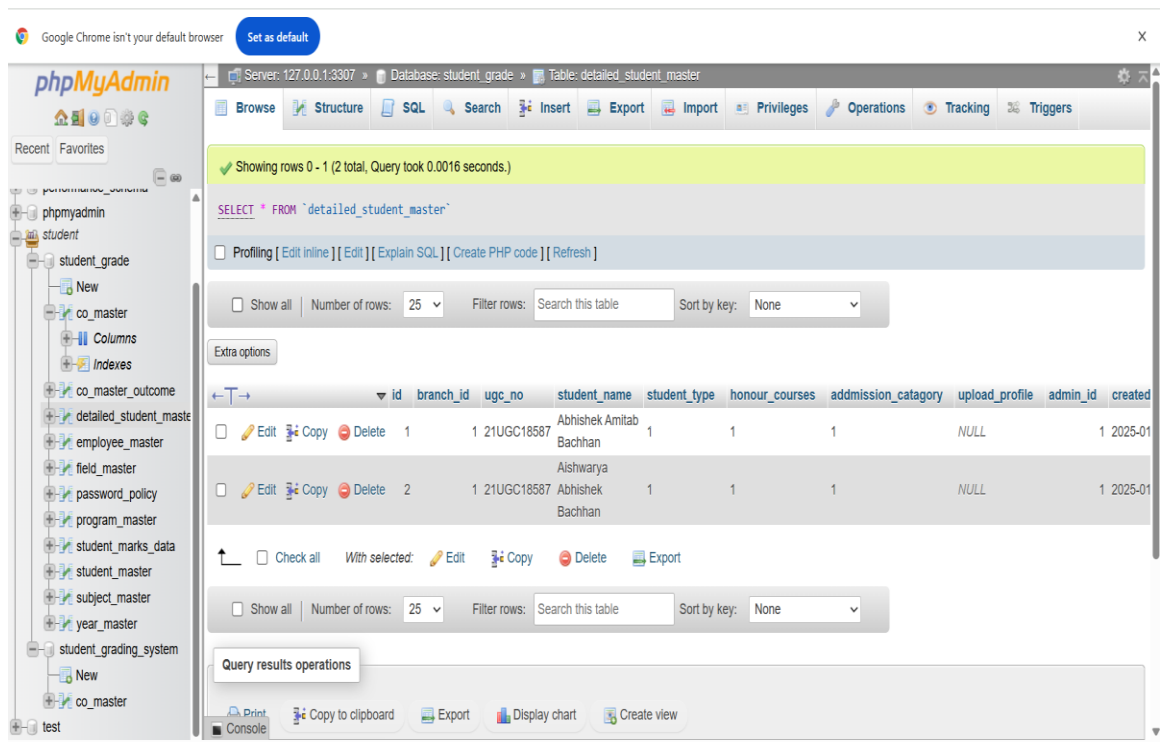
- Faculty enters student marks (internal assessments, final exams, etc.).
- Performance metrics are recorded.

## 12. Upload Indirect Feedback

- Faculty inputs feedback from students, alumni, and employers.
- Indirect measures of learning effectiveness are collected.

## 13. Attainment

- The system evaluates attainment levels by comparing actual student performance against target COs, POs, and PEOs.
- Used for continuous improvement and accreditation purposes.



The phpMyAdmin screenshot displays the "detailed\_student\_master" table within the "student\_grade" database. It contains information about individual students, including their branch, UGC number, name, and admission details.

Showing rows 0 - 5 (6 total, Query took 0.0002 seconds.)

```
SELECT * FROM `student_marks_data`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	exam_id	student_id	subject_id	co_1	co_2	co_3	co_4	co_5	co_6	total	admin_id	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	68	2	12	15	14	0	0	0	41	156	2025-03-22 10:37:40	2025-03-22 10:37
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	69	2	5	0	3	14	0	0	22	156	2025-03-22 10:38:23	2025-03-22 10:38
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	69	2	5	7	9	8	7	10	46	156	2025-03-22 10:39:03	2025-03-22 10:39
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	1	69	3	15	10	5	0	0	0	30	157	2025-03-22 10:46:10	2025-03-22 10:46
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	2	69	3	7	8	9	0	0	0	24	157	2025-03-22 10:46:52	2025-03-22 10:46
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	3	10	7	6	8	9	5	45			157	2025-03-22 10:47:33	2025-03-22 10:47

This shows the "student\_marks\_data" table within the "student\_grade" database. It contains records of student exam scores for subject ID 69, broken down into six CO (Course Outcome) components, along with total marks and administrative information.

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT * FROM `program_master`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	name	credits	is_active	admin_id	created_at	updated_at	deleted_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	UG	96	1	155	2024-08-21 10:50:32	2024-11-09 04:15:13	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	PG	10	1	155	2024-08-21 10:50:32	2024-11-09 03:12:18	NULL

The screenshot displays the "program\_master" table in the "student\_grade" database using phpMyAdmin. It shows two program records: one for "UG" with 96 credits and another for "PG" with 10 credits, along with their active status and timestamps.

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

```
SELECT * FROM `student_marks_data`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	exam_id	student_id	subject_id	co_1	co_2	co_3	co_4	co_5	co_6	total	admin_id	created_at	updated_at
<input type="checkbox"/>	1	1	68	2	12	15	14	0	0	0	41	156	2025-03-22 10:37:40	2025-03-22 10:37
<input type="checkbox"/>	2	2	69	2	5	0	3	14	0	0	22	156	2025-03-22 10:38:23	2025-03-22 10:38
<input type="checkbox"/>	3	3	69	2	5	7	9	8	7	10	46	156	2025-03-22 10:39:03	2025-03-22 10:39
<input type="checkbox"/>	4	1	69	3	15	10	5	0	0	0	30	157	2025-03-22 10:46:10	2025-03-22 10:46
<input type="checkbox"/>	5	2	69	3	7	8	9	0	0	0	24	157	2025-03-22 10:46:52	2025-03-22 10:46
<input type="checkbox"/>	6	3	69	3	10	7	6	8	9	5	45	157	2025-03-22 10:47:33	2025-03-22 10:47

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

The screenshot displays the "student\_marks\_data" table within the "student\_grade" database in phpMyAdmin. It shows student exam scores for various subjects, detailed by CO components, total marks, and administrative data.

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

```
SELECT * FROM `subject_master`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	sem_id	teacher_id	program_type_id	year_id	field_code	subject_type_id	subject_code	semis_id	name	credits
<input type="checkbox"/>	1	3	156	1	1	1	1	CSE101	4	DATA SCIENCE	20
<input type="checkbox"/>	2	3	156	1	1	1	1	CSE02	4	ARTIFICIAL INTELLIGENCE	20
<input type="checkbox"/>	3	1	157	1	1	1	1	CSE201	4	PYTHON	20
<input type="checkbox"/>	4	1	157	1	1	1	1	CSE202	4	C++	20

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

This phpMyAdmin view shows the "student\_marks\_data" table from the "student\_grade" database. It presents records of student marks for different exams and subjects, broken down into individual CO scores and a total.

## Chapter 6

### Testing

---

#### 6.1 Fundamentals of Testing

- **Testing:** Testing is the process of evaluating a system or software to identify any discrepancies between actual and expected results. It is performed to ensure that the software meets the specified requirements and works as intended.
- **Purpose of Testing:** The main purpose of testing is to uncover defects or bugs in the software and provide confidence in its quality. It helps identify areas for improvement and ensures that the software is reliable, functional, and meets user expectations.
- **Testing Objectives:** The primary objectives of testing include verifying the correctness and completeness of the software, validating that it meets the user requirements, detecting defects and errors, and ensuring the software's reliability and robustness.
- **Testing Techniques:** Various testing techniques are used to ensure comprehensive coverage of the software. These techniques include unit testing, integration testing, system testing, acceptance testing, regression testing, and performance testing. Each technique focuses on different aspects of the software and helps identify different types of defects.
- **Test Plan:** A test plan is a document that outlines the testing approach, objectives, scope, and schedule. It defines the test strategy, test cases, and test environment. The test plan serves as a roadmap for the testing process, ensuring that all necessary aspects are covered.
- **Test Cases:** Test cases are specific scenarios or conditions that are designed to test the functionality of the software. They consist of inputs, expected outputs, and

preconditions. Test cases are created based on the requirements and specifications of the software to ensure comprehensive testing coverage

- **Defect Reporting:** When defects or issues are identified during testing, they are documented and reported. A defect report typically includes information about the defect, such as its description, steps to reproduce, severity, and priority. Defects are then tracked and managed using a defect tracking system.
- **Test Execution:** Test execution involves running the test cases on the software and comparing the actual results with the expected results. It helps identify any discrepancies or defects in the software. Test execution is typically performed in a controlled testing environment to ensure accurate and reliable results.
- **Test Documentation:** Throughout the testing process, documentation plays a crucial role. It includes test plans, test cases, test results, defect reports, and any other relevant information. Test documentation helps in maintaining a record of the testing activities, tracking progress, and facilitating communication among the testing team and stakeholders.
- **Test Completion and Evaluation:** After executing all the planned tests and addressing the identified defects, the testing process is considered complete. A final evaluation is conducted to assess the overall quality of the software and determine if it meets the defined acceptance criteria. The testing team provides recommendations and feedback for further improvements.



## 6.2 Test Plan of the Project

### 1. Introduction

- Overview of the project
- Purpose and objectives of the test plan
- Scope and limitations of testing

### 2. Test Strategy

- Test levels (unit testing, integration testing, system testing, etc.)
- Test types (functional, non-functional, regression, etc.)
- Test environment and infrastructure requirements

### 3. Test Deliverables

- List of documents and artifacts to be produced during testing
- Test cases, test scripts, and test data
- Test reports, defect reports, and other testing documentation

### 4. Test Schedule

- Timeline for test preparation, execution, and completion
- Milestones and deadlines for each testing phase
- Resource allocation for testing activities

### 5. Test Environment

- Hardware and software requirements for the test environment
- Configuration management of test environments
- Test data and test data base setup

6. Test Execution

- Test case prioritization and sequencing
- Test execution approach(manual, automated, or a combination)
- Test data management and test coverage analysis
- Test execution schedule and responsibilities

7. Test Evaluation

- Criteria for evaluating test results
- Defect reporting and management process
- Metrics and measurements for assessing testing progress and effectiveness -  
Criteria for determining test completion and exit criteria

8. Risks and Assumptions

- Identification of potential risks and their impact on testing
- Mitigation strategies for identified risks
- Assumptions made during testing

9. Test Team and Responsibilities

- Roles and responsibilities of the testing team members
- Communication and coordination channels
- Training and skill development requirements

10. Approval and Sign-Off

- Sign-off process for test plan approval
- Stakeholder and project manager approval

### 6.3. Test Cases

#### Positive Test Cases

##### Test Cases for Principal

Test Case ID	Description	Input	Expected Output
TC_P_01	Create vision and mission	Vision and mission text	Vision and mission saved
TC_P_02	Create UG/PG program	Program name, type	Program created successfully
TC_P_03	Assign HOD to a program	Program ID, HOD ID	HOD assigned to program

##### Test Cases for Dean

Test Case ID	Description	Input	Expected Output
TC_D_01	Assign credits to UG/PG	Program ID, credit value	Credits assigned
TC_D_02	Assign credits to department	Department ID, credit value	Department credits updated
TC_D_03	Assign credits to semester	Semester number, credit value	Semester credits updated
TC_D_04	Format question papers	Course ID, type (ISE/ESE)	Format applied

**Test Cases for HOD**

<b>Test Case ID</b>	<b>Description</b>	<b>Input</b>	<b>Expected Output</b>
TC_H_01	Set department vision & mission	Text input	Saved successfully
TC_H_02	Add semesters and define courses	Course list, semester mapping	Semester structure created
TC_H_03	Upload POs, PEOs, PSOs	Outcome definitions	Outcomes uploaded
TC_H_04	Define and upload CO-PO mapping	COs, PO links	Mapping saved
TC_H_05	Assign course to faculty	Course ID, Faculty ID	Course assigned
TC_H_06	Map CO with PO, PSO, PEO	Mapping matrix	Saved and validated

**Test Cases for Faculty**

<b>Test Case ID</b>	<b>Description</b>	<b>Input</b>	<b>Expected Output</b>
TC_F_01	Upload curriculum (theory & practical)	Curriculum files	Files uploaded successfully
TC_F_02	Upload marks (ISE, ESE, POE, TW, etc.)	Student scores	Marks recorded
TC_F_03	Upload extracurricular activities	Event details, student ID	Activities recorded
TC_F_04	Map student attainment (direct/indirect)	Mapping matrix	Attainment calculated

**Test Cases for Admin User**

<b>Test Case ID</b>	<b>Description</b>	<b>Input</b>	<b>Expected Output</b>
TC_A_01	Monitor system status	N/A	Status report generated
TC_A_02	Manage user access	User role, permissions	Access updated

**Admin User Login Test Cases**

<b>TC ID</b>	<b>Test Case Description</b>	<b>Input</b>	<b>Expected Result</b>
TC_AD_01	Login as Admin	Valid credentials	Admin dashboard loads
TC_AD_02	Monitor system operations	N/A	System status displayed
TC_AD_03	Verify access control	User role update	Role/permissions updated
TC_AD_04	Manage users	Add/remove/edit user	User database updated
TC_AD_05	System performance check	N/A	Logs, usage, uptime visible

**Principal Login Test Cases**

TC ID	Test Case Description	Input	Expected Result
TC_PR_01	Login as Principal	Valid credentials	Principal dashboard loads
TC_PR_02	Define vision and mission	Text input	Vision and mission saved
TC_PR_03	Create UG/PG program	Program name, type	Program successfully created
TC_PR_04	Assign HOD to program	HOD ID, program ID	HOD assignment successful
TC_PR_05	Validate role access	N/A	Only authorized features accessible

**Dean Login Test Cases**

TC ID	Test Case Description	Input	Expected Result
TC_DE_01	Login as Dean	Valid credentials	Dean dashboard loads
TC_DE_02	Assign UG/PG credits	Program ID, credits	Credits assigned
TC_DE_03	Assign credits to department	Department ID, value	Credits updated
TC_DE_04	Assign semester-wise credits	Semester info, credits	Semester updated
TC_DE_05	Format question paper	Course ID, exam type	Format applied successfully

**HOD Login Test Cases**

TC ID	Test Case Description	Input	Expected Result
TC_HOD_01	Login as HOD	Valid credentials	HOD dashboard loads
TC_HOD_02	Define department vision and mission	Text input	Data saved
TC_HOD_03	Add semesters and define courses	Semester, course data	Curriculum structure created
TC_HOD_04	Upload PO, PEO, PSO	Outcome text/data	Outcomes uploaded
TC_HOD_05	Upload CO-PO mappings	CO-PO matrix	Mapping saved
TC_HOD_06	Assign courses to faculty	Course ID, Faculty ID	Assignment successful
TC_HOD_07	Map COs to POs/PEOs/PSOs	Mapping matrix	Mapping stored and linked

**Faculty Login Test Cases**

TC ID	Test Case Description	Input	Expected Result
TC_FA_01	Login as Faculty	Valid credentials	Faculty dashboard loads
TC_FA_02	Upload curriculum (theory/practical)	Files/documents	Upload successful
TC_FA_03	Upload marks (ISE, ESE, etc.)	Student marks	Marks recorded
TC_FA_04	Upload extracurricular data	Event/activity info	Activity stored
TC_FA_05	Conduct attainment mapping	Student data, mapping	Attainment calculated and displayed

### Negative Test Cases

#### Test Cases for Principal

Test Case ID	Description	Input	Expected Output
TC_P_01_N1	Attempt to create vision with empty text	Vision and mission text left blank	Error message: "Vision text cannot be empty"
TC_P_01_N2	Attempt to create mission with excessive length	Mission text exceeding allowed characters	Error message: "Mission text exceeds the maximum allowed length"
TC_P_02_N1	Attempt to create program with empty name	Program name left blank	Error message: "Program name is required"
TC_P_02_N2	Attempt to create program with invalid type	Program type: "Diploma"	Error message: "Invalid program type"
TC_P_02_N3	Attempt to create program with existing name	Program name already in the system	Error message: "Program with this name already exists"
TC_P_03_N1	Attempt to assign non-existent HOD	Program ID: 1, HOD ID: 999	Error message: "Invalid HOD ID"
TC_P_03_N2	Attempt to assign HOD to non-existent program	Program ID: 999, HOD ID: 1	Error message: "Invalid Program ID"



**Test Cases for Dean**

Test Case ID	Description	Input	Expected Output
TC_D_01_N1	Attempt to assign negative credits to program	Program ID: 1, credit value: -5	Error message: "Credit value cannot be negative"
TC_D_01_N2	Attempt to assign credits to non-existent program	Program ID: 999, credit value: 100	Error message: "Invalid Program ID"
TC_D_01_N3	Attempt to assign zero credits to program	Program ID: 1, credit value: 0	Warning message: "Are you sure you want to assign zero credits?"
TC_D_02_N1	Attempt to assign negative credits to department	Department ID: 1, credit value: -10	Error message: "Credit value cannot be negative"
TC_D_02_N2	Attempt to assign credits to non-existent department	Department ID: 999, credit value: 50	Error message: "Invalid Department ID"
TC_D_03_N1	Attempt to assign negative credits to semester	Semester number: 1, credit value: -2	Error message: "Credit value cannot be negative"
TC_D_03_N2	Attempt to assign credits to invalid semester	Semester number: 10, credit value: 15	Error message: "Invalid Semester Number"
TC_D_04_N1	Attempt to format question paper with invalid course ID	Course ID: 999, type: "ISE"	Error message: "Invalid Course ID"
TC_D_04_N2	Attempt to format question paper with invalid type	Course ID: 1, type: "Midterm"	Error message: "Invalid question paper type"

**Test Cases for HOD**

Test Case ID	Description	Input	Expected Output
TC_H_01_N1	Attempt to set department vision with empty text	Text input (vision): ""	Error message: "Department vision cannot be empty"
TC_H_01_N2	Attempt to set department mission with excessive length	Text input (mission) exceeding allowed characters	Error message: "Department mission exceeds the maximum allowed length"
TC_H_02_N1	Attempt to add semester with invalid number	Course list: [...], semester mapping: {0: [...]}	Error message: "Invalid semester number"
TC_H_02_N2	Attempt to define course without assigning to a semester	Course list: ["New Course"], semester mapping: {}	Warning message: "Course 'New Course' is not assigned to any semester"
TC_H_03_N1	Attempt to upload POs with incorrect file format	Outcome definitions: "POs.txt"	Error message: "Invalid file format for POs. Please upload a valid file"
TC_H_03_N2	Attempt to upload PEOs with empty file	Outcome definitions: (empty file)	Error message: "PEOs file cannot be empty"
TC_H_04_N1	Attempt to define CO-PO mapping with invalid CO	COs: [...], PO links: {"CO7": ["PO1"]}	Error message: "Invalid CO ID: CO7"
TC_H_04_N2	Attempt to define CO-PO mapping with invalid PO	COs: [...], PO links: {"CO1": ["PO15"]}	Error message: "Invalid PO ID: PO15"

**Test Cases for Faculty**

Test Case ID	Description	Input	Expected Output
TC_F_01_N1	Attempt to upload curriculum with incorrect file type	Curriculum files: "syllabus.doc"	Error message: "Invalid file type for curriculum"
TC_F_01_N2	Attempt to upload empty curriculum file	Curriculum files: (empty file)	Error message: "Curriculum file cannot be empty"
TC_F_02_N1	Attempt to upload marks for non-existent student	Student scores: {"999": {"ISE": 25}}	Error message: "Invalid Student ID: 999"
TC_F_02_N2	Attempt to upload marks with invalid score value	Student scores: {"1": {"ESE": 120}}	Error message: "Invalid score for ESE. Score exceeds maximum"
TC_F_02_N3	Attempt to upload marks for an invalid assessment type	Student scores: {"1": {"Quiz": 10}}	Error message: "Invalid assessment type: Quiz"
TC_F_03_N1	Attempt to upload extracurricular activity for non-existent student	Event details: {...}, Student ID: 999	Error message: "Invalid Student ID: 999"
TC_F_03_N2	Attempt to upload extracurricular activity with missing details	Event details: {"name": "Event A"}	Error message: "Event details are incomplete"
TC_F_04_N1	Attempt to map attainment for non-existent student	Mapping matrix: {"999": {"direct": 80}}	Error message: "Invalid Student ID: 999"
TC_F_04_N2	Attempt to map attainment with invalid mapping type	Mapping matrix: {"1": {"external": 75}}	Error message: "Invalid attainment mapping type: external"

**Test Cases for Admin User**

Test Case ID	Description	Input	Expected Output
TC_A_01_N1	Attempt to monitor system with insufficient permissions	N/A	Error message: "Access denied. Insufficient permissions."
TC_A_02_N1	Attempt to manage user with invalid role	User role: "Guest", permissions: [...]	Error message: "Invalid user role"
TC_A_02_N2	Attempt to update access for non-existent user	User role: "Student", ID: 999	Error message: "User not found"

**Login test cases**

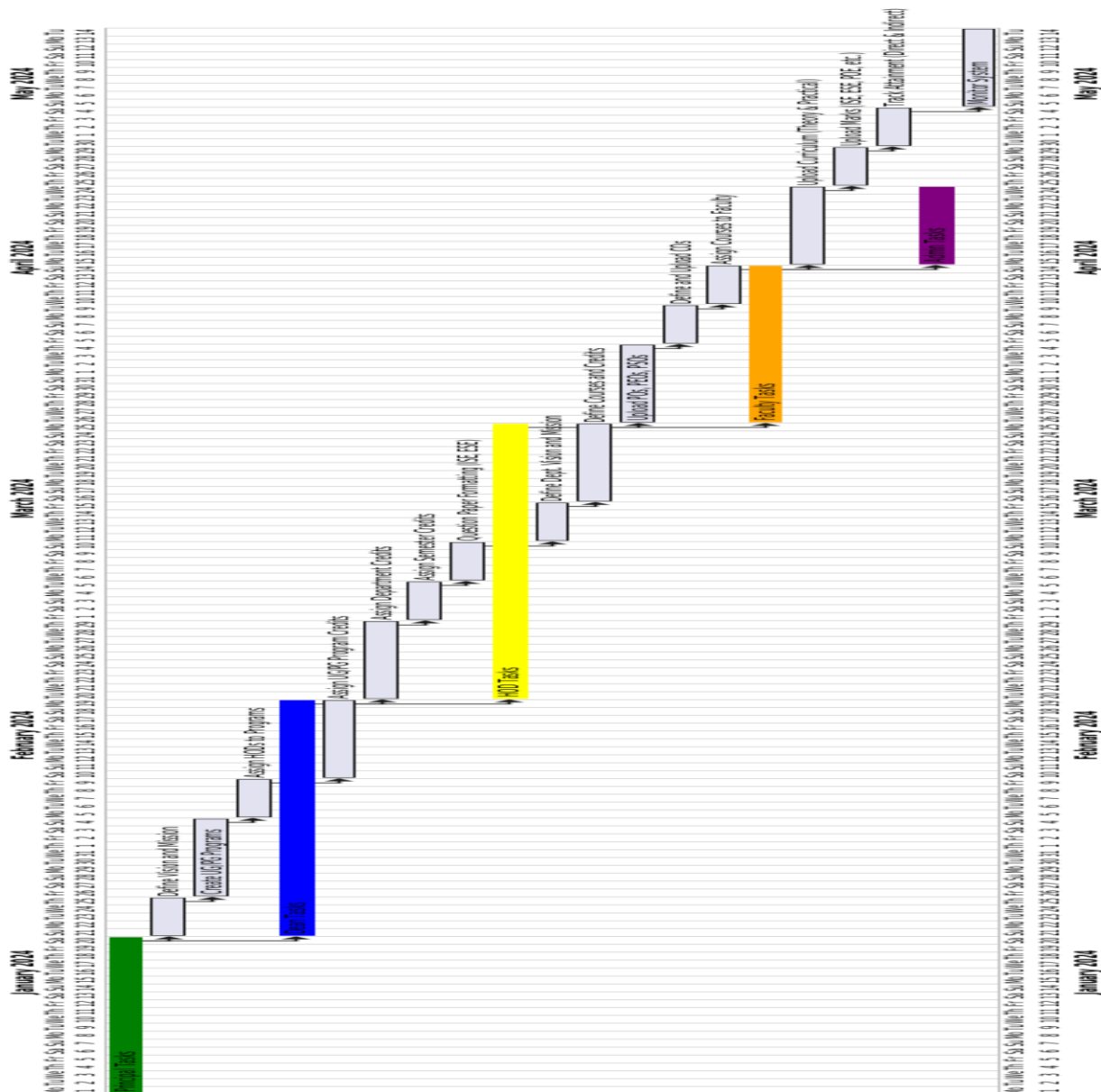
TC ID	User Role	Test Case Description	Input	Expected Result
TC_AD_01	Admin	Login as Admin with valid credentials	Username: "validadmin", Password: "validpassword"	Admin dashboard loads
TC_AD_01_N1	Admin	Login as Admin with invalid username	Username: "wrongadmin", Password: "validpassword"	Error message: "Invalid username or password"
TC_AD_01_N2	Admin	Login as Admin with invalid password	Username: "validadmin", Password: "wrongpassword"	Error message: "Invalid username or password"

TC_AD_01_N3	Admin	Login as Admin with empty credentials	Username: "", Password: ""	Error message: "Username and password are required"
TC_PR_01	Principal	Login as Principal with valid credentials	Username: "validprincipal", Password: "valid"	Principal dashboard loads
TC_PR_01_N1	Principal	Login as Principal with invalid username	Username: "wrongprincipal", Password: "valid"	Error message: "Invalid username or password"
TC_PR_01_N2	Principal	Login as Principal with invalid password	Username: "validprincipal", Password: "wrong"	Error message: "Invalid username or password"
TC_DE_01	Dean	Login as Dean with valid credentials	Username: "validdean", Password: "good"	Dean dashboard loads
TC_DE_01_N1	Dean	Login as Dean with invalid credentials	Username: "wrongdean", Password: "bad"	Error message: "Invalid username or password"

## Chapter 7

### Project Plan & Schedule

#### 7.1 Gantt Chart



<b>Task Name</b>	<b>Start Date</b>	<b>End Date</b>	<b>Duration</b>
Initiation	Jan 01, 2004	Jan 07, 2004	7 days
Scope Definition	Jan 08, 2004	Jan 14, 2004	7 days
Initial Planning	Jan 15, 2004	Jan 23, 2004	9 days
WBS Development	Jan 24, 2004	Feb 04, 2004	12 days
Risk Management	Feb 05, 2004	Feb 11, 2004	7 days
Budget Development	Feb 12, 2004	Feb 22, 2004	11 days
Baseline Finalization (MS #1)	Feb 23, 2004	Feb 23, 2004	1 day
Procurement Planning	Feb 24, 2004	Mar 01, 2004	7 days
Supplier Solicitation	Mar 02, 2004	Mar 08, 2004	7 days
Select Provider (MS #2)	Mar 09, 2004	Mar 09, 2004	1 day
Joint Dev. and Test Planning	Mar 10, 2004	Mar 20, 2004	11 days
Update Dev. Req's (FRS, SRS)	Mar 21, 2004	Mar 26, 2004	6 days
Develop Code & Initial Tests	Mar 27, 2004	Apr 07, 2004	12 days
Alpha Testing	Apr 08, 2004	Apr 16, 2004	9 days
Update Code, FRS/SRS	Apr 17, 2004	Apr 23, 2004	7 days
Develop Training Materials	Apr 24, 2004	Apr 29, 2004	6 days
Conduct Training	Apr 30, 2004	May 07, 2004	8 days
Go Live (MS #3)	May 08, 2004	May 08, 2004	1 day

## Chapter 8

# Risk Management and Analysis

---

### 8.1 Risk Analysis

1. Failure of the monitoring system:
  - Likelihood: Medium
  - Impact: Poor assessment of program effectiveness, risking accreditation.
  - Mitigation: Conduct thorough system testing and regular audits.
2. Errors in credit assignment:
  - Likelihood: High
  - Impact: Discrepancies in academic programs, causing student dissatisfaction.
  - Mitigation: Use automated systems with double-verification processes.
3. Miscommunication between key actors:
  - Likelihood: Medium
  - Impact: Misaligned decisions, inefficiencies, and delays.
  - Mitigation: Establish structured communication workflows.
4. Improper delegation of responsibilities:
  - Likelihood: Medium
  - Impact: Delays and quality issues in program execution.
  - Mitigation: Clearly define roles and responsibilities and provide training.
5. Misalignment of vision and mission:
  - Likelihood: Low
  - Impact: Non-compliance with accreditation standards.



- Mitigation: Involve all stakeholders during planning and conduct periodic reviews.
6. Errors in question paper formatting:
- Likelihood: Medium
  - Impact: Inequitable or unclear evaluations for students.
  - Mitigation: Implement automated formatting tools and quality assurance checks.
7. Unauthorized access to sensitive data:
- Likelihood: High
  - Impact: Breach of confidentiality and reputational damage.
  - Mitigation: Enforce strong access controls and encryption protocols.
8. Delays in implementing new programs:
- Likelihood: Medium
  - Impact: Failure to meet academic timelines, affecting reputation.

## Chapter 9

# Configuration Management

---

### 9.1 Installation /Uninstallation

#### 1. Software Installation

Install XAMPP (Apache, MySQL, PHP):

- Go to <https://www.apachefriends.org/index.html>
- Download the latest version of XAMPP for your OS (Windows/Linux/macOS)
- Run the installer and complete installation with default settings
- Open XAMPP Control Panel
- Start the Apache and MySQL services

Verify Installation:

- Open a browser and go to <http://localhost> — the XAMPP dashboard should appear
- Visit <http://localhost/phpmyadmin> — phpMyAdmin should open if MySQL is running

#### 2. Configuration

Apache Server:

- Apache is preconfigured for PHP with XAMPP, no extra setup needed

MySQL Database:

- Go to <http://localhost/phpmyadmin>
- Click "New" to create a new database, name it OBE\_System
- Inside the OBE\_System database, create tables like COs, POs, and PEOs using SQL or GUI

Example SQL Table for COs:

sql  
CopyEdit  
CREATE TABLE COs (  
    id INT AUTO\_INCREMENT PRIMARY KEY,  
    course\_code VARCHAR(20),  
    co\_description TEXT  
);

PHP and MySQL Connection:

- Use this sample PHP script to connect:

php  
CopyEdit  
<?php  
\$conn = new mysqli("localhost", "root", "", "OBE\_System");  
if (\$conn->connect\_error) {  
    die("Connection failed: " . \$conn->connect\_error);  
}  
echo "Connection successful!";  
?>

- Save the file as test\_connection.php in C:\xampp\htdocs and access it at [http://localhost/test\\_connection.php](http://localhost/test_connection.php)

### 3. Connection Between Components

- Ensure PHP can query MySQL with mysqli
- Example script to retrieve COs:

php

CopyEdit

```
<?php
```

```
$conn = new mysqli("localhost", "root", "", "OBE_System");
```

```
$sql = "SELECT * FROM COs";
```

```
$result = $conn->query($sql);
```

```
while($row = $result->fetch_assoc()) {
```

```
    echo "CO: " . $row["co_description"] . "<br>";
```

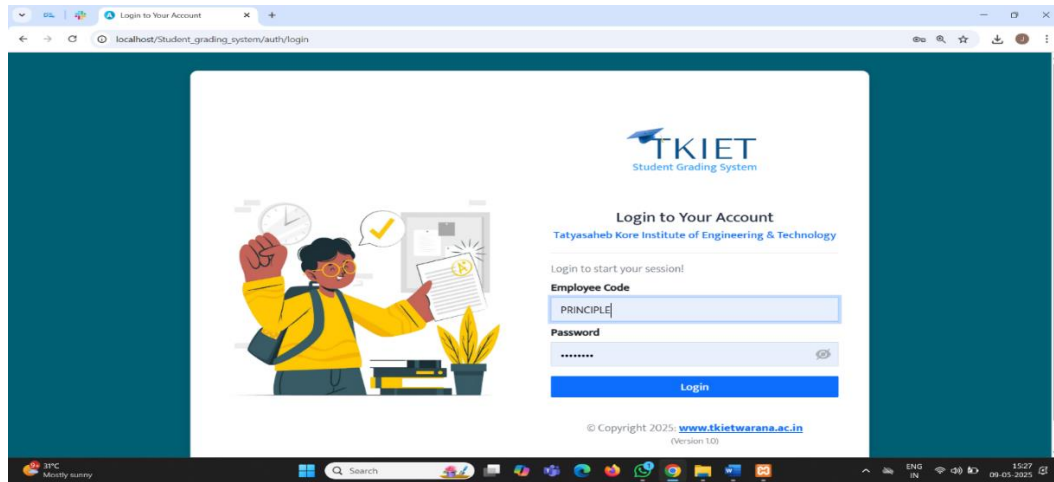
```
}
```

```
$conn->close();
```

```
?>
```

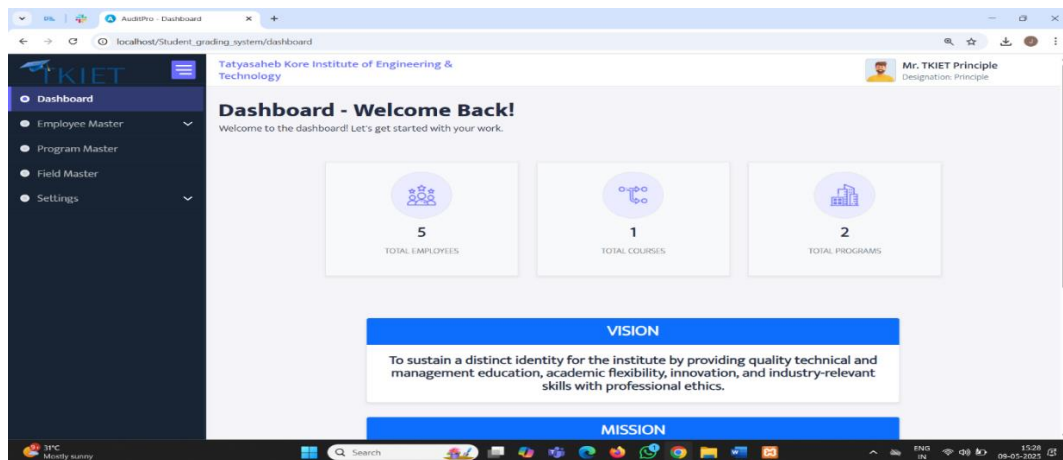
## 9.2 User Manual

### 9.2.1 Login Page



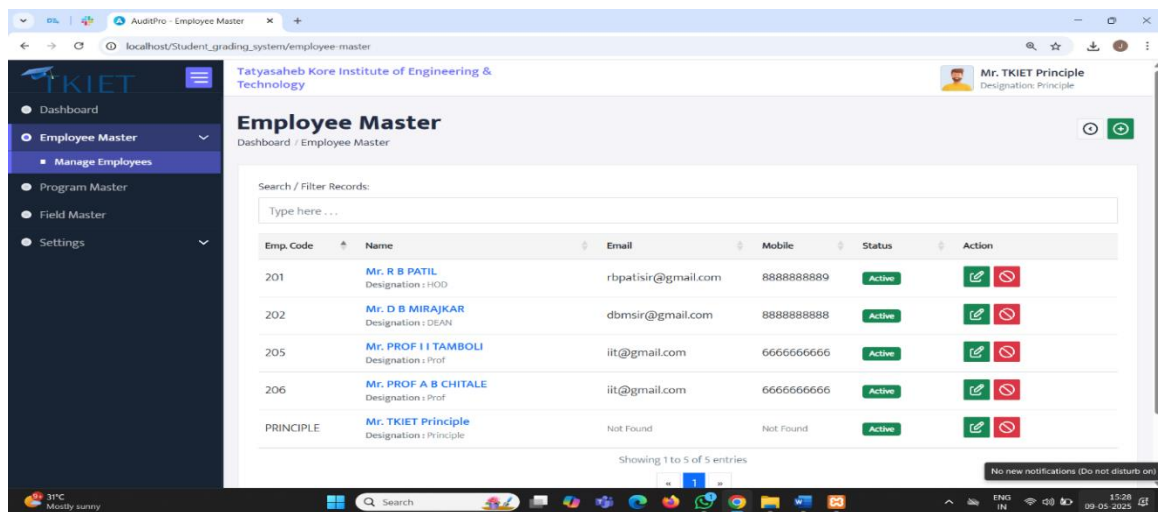
The page for the TKIET Student Grading System. It prompts users to enter their Employee Code and Password to access their account.

### 9.2.2 Principle Home Page



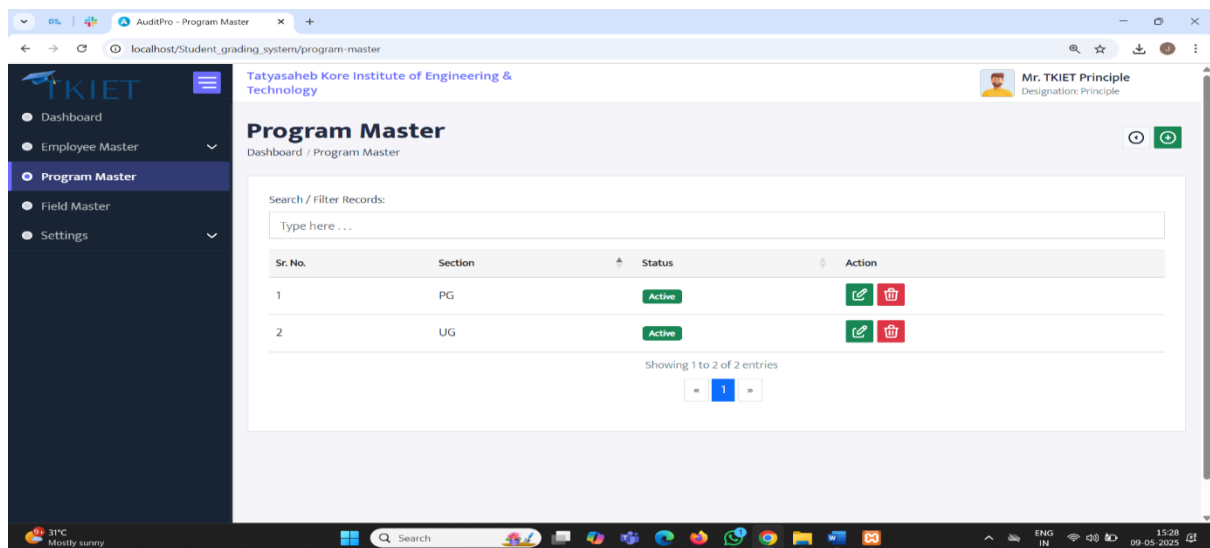
The TKIET Student Grading System dashboard welcomes "Mr. TKIET Principle" and presents key metrics: 5 employees, 1 course, and 2 programs. It also displays the institution's vision and mission.

### 9.2.3 Employee Master



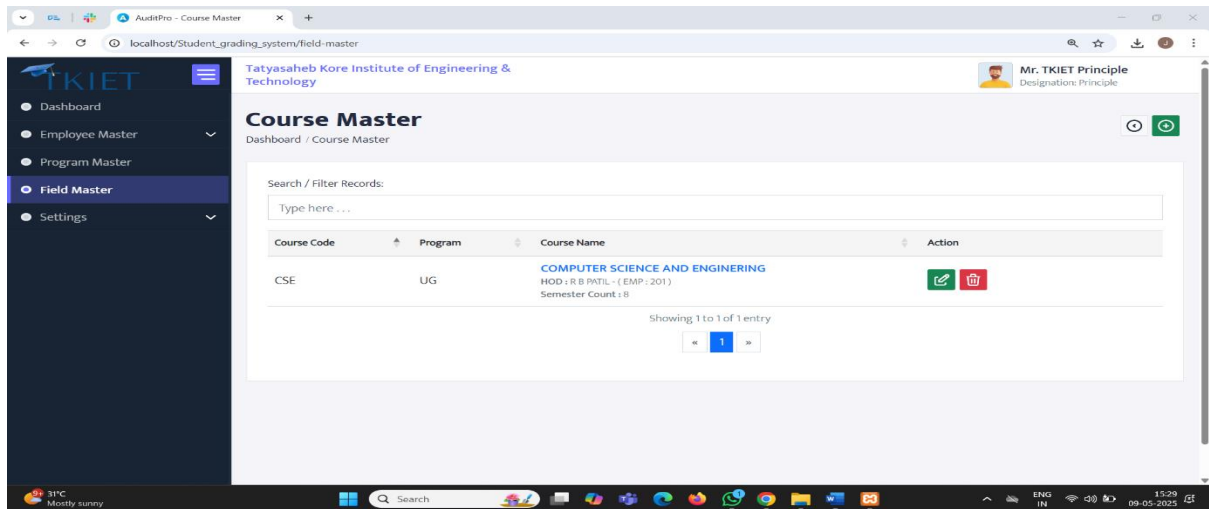
The TKIET Student Grading System's "Employee Master" lists employees with their code, name, designation, contact details, and status. Actions to edit/view and deactivate each employee are also available.

### 9.2.4 Program Master



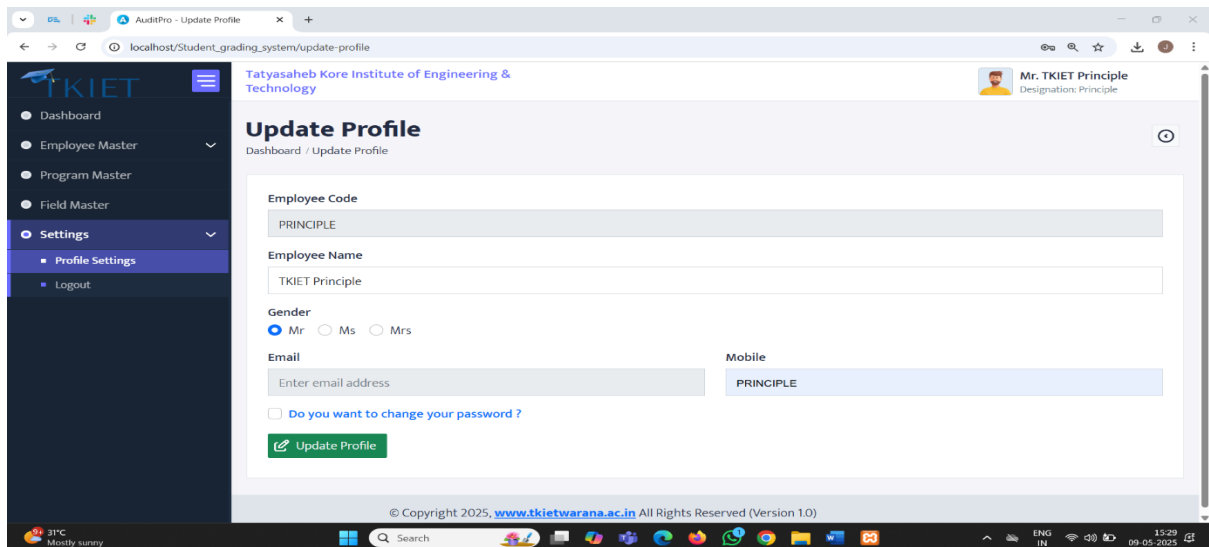
The TKIET system's "Program Master" lists academic programs like "PG" and "UG," both currently "Active." Options to edit/view and delete each program are provided.

### 9.2.5 Course Master



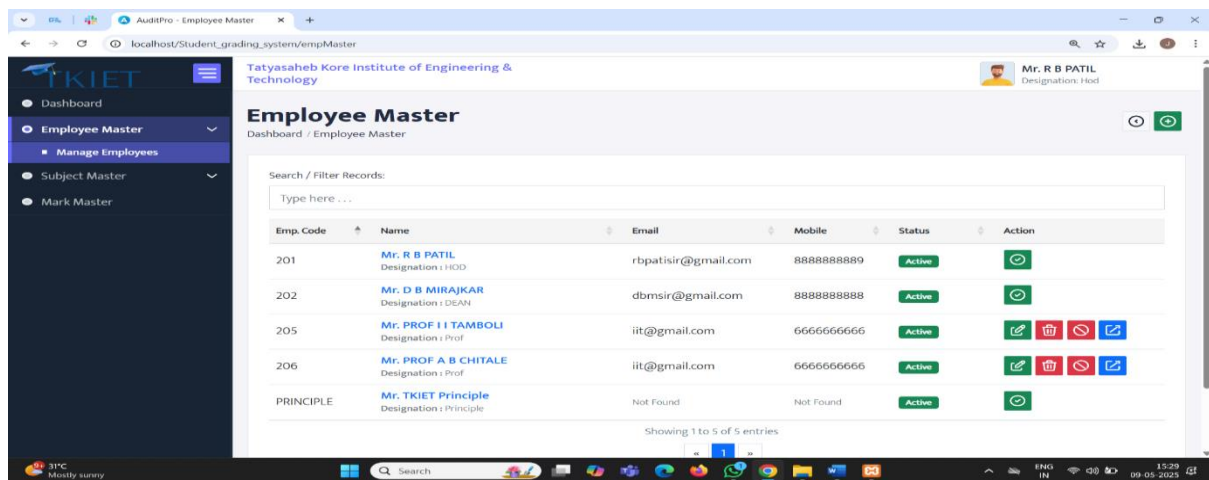
The TKIET system's "Course Master" lists courses like "COMPUTER SCIENCE AND ENGINEERING" (under "UG" with code "CSE"). Options to edit/view and delete the course are available.

### 9.2.6 Update Profile



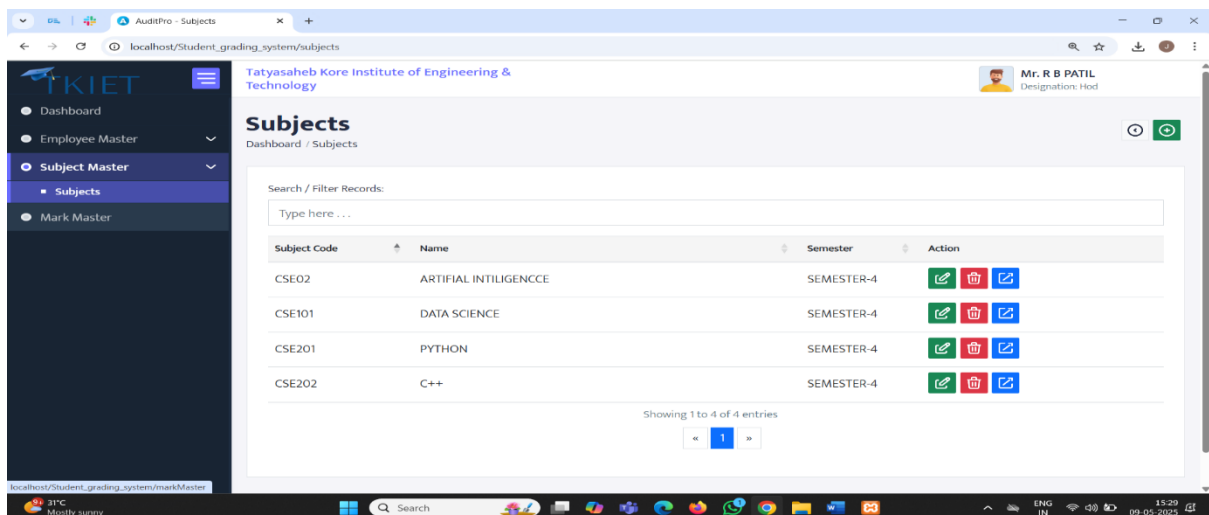
The TKIET system's "Update Profile" allows "Mr. TKIET Principle" to modify his name, gender, email, and mobile number. It also provides an option to change the account password.

### 9.2.7 Employee Master (HOD)



The TKIET system's "Employee Master" displays a list of employees, including their code, name, designation, contact details, and status. Logged in as "Mr. R B PATIL (Hod)," the system offers options to view, edit, and deactivate employee accounts.

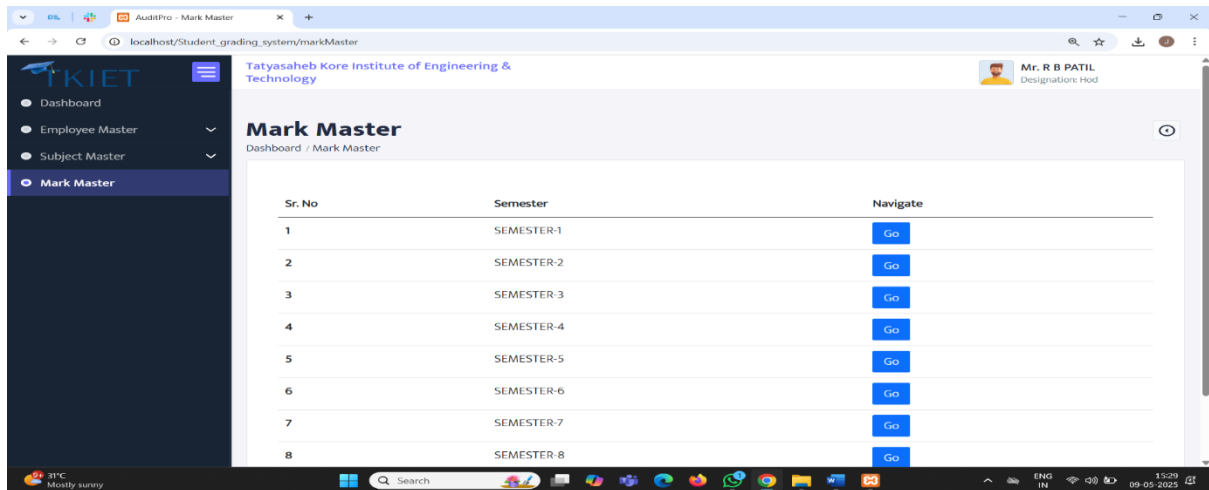
### 9.2.8 Subject Master



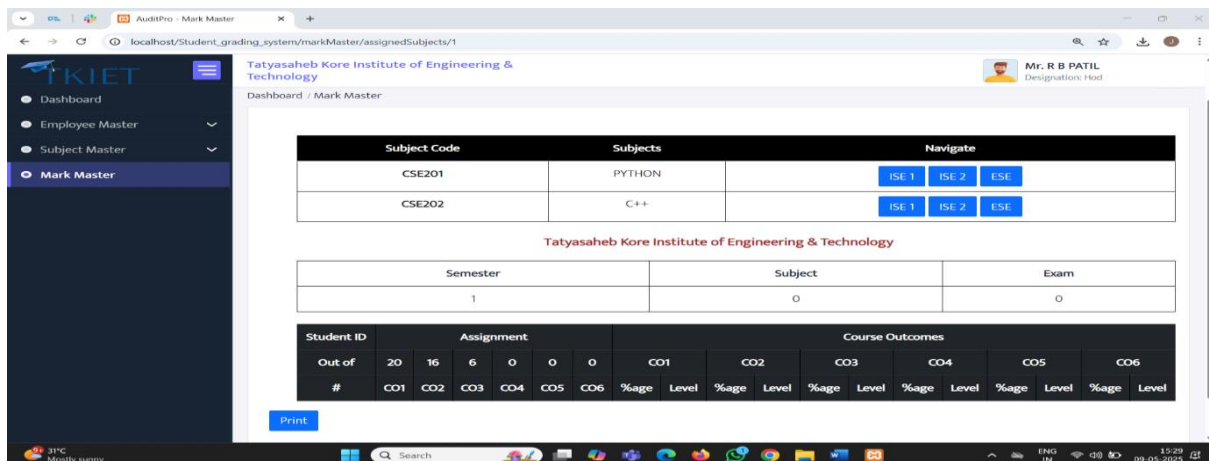
The TKIET system's "Subjects" lists courses like "ARTIFICIAL INTELLIGENCE," "DATA SCIENCE," "PYTHON," and "C++" for SEMESTER-4, along with their codes. Logged in as "Mr. R B PATIL (Hod)," the system provides options to view, edit, and map each subject.



### 9.2.9 Mark Master



The TKIET system's "Mark Master" lists all eight semesters with a "Go" button for each. Logged in as "Mr. R B PATIL (Hod)," this section likely allows navigation to manage marks for each semester.



The TKIET "Mark Master" for Semester 1 shows "PYTHON" and "C++" with ISE 1, ISE 2, and ESE navigation. It also provides sections for assignment marks and Course Outcome entry, along with student-specific input fields. Logged in as "Mr. R B PATIL (Hod)."

### 9.2.10 Subject Manager (Faculty)

**Add Educational Details**

Student Roll No  
Please select student

CO1  
Please enter CO1

CO2  
Please enter CO2

CO3  
Please enter CO3

CO4  
Please enter CO4

CO5  
Please enter CO5

CO6  
Please enter CO6

**Add Marks**

Student Name	Subject	CO1	CO2	CO3	CO4	CO5	CO6	total
69	2	5	0	3	14	0	0	22

Showing 1 to 1 of 1 entry (filtered from 6 total entries)

TKIET's "Add Educational Details" for a subject, with tabs for internal and external exams and CO mark entry. Existing marks are shown in a table.

**Subject**  
Dashboard / Subject

Search / Filter Records:  
Type here ...

Subject Code	Name	Semester	Action
CSE02	ARTIFICIAL INTELLIGENCE	SEMESTER-4	
CSE101	DATA SCIENCE	SEMESTER-4	

Showing 1 to 2 of 2 entries (filtered from 4 total entries)

TKIET's "Subject" section lists courses ("ARTIFICIAL INTELLIGENCE," "DATA SCIENCE" for SEMESTER-4) with an action button. Faculty user is "Mr. PROF I I TAMBOLI."

## **Chapter 10**

### **Conclusion and Future Scope**

---

#### **10.1 Conclusion**

The implementation of Outcome-Based Education (OBE) represents a significant step toward enhancing the quality and relevance of engineering education by aligning learning processes with specific, measurable outcomes. This project has explored the challenges faced by faculty in adopting OBE, including issues in defining learning outcomes, aligning curriculum and assessments, and managing increased workloads. The proposed online expert system provides a structured and efficient solution to these challenges, enabling role-based management of Course Outcomes (COs) and Program Outcomes (POs), and supporting continuous feedback and improvement.

By integrating this system, educational institutions can streamline OBE processes, promote accountability, and ensure that graduates acquire industry-relevant skills and competencies. The system not only supports the vision of academic excellence and institutional growth but also upholds the core principles of equity, inclusivity, and continuous improvement. Ultimately, this initiative empowers educators and students alike, contributing to the creation of a robust, future-ready educational ecosystem that drives both individual and societal progress.

#### **10.2 Future Scope**

The Outcome-Based Education (OBE) Management System lays a strong foundation for streamlining academic processes and aligning educational activities with defined

learning outcomes. Building upon the current functionalities, several enhancements can be explored to further improve system efficiency, scalability, and impact:

### **Integration with External Systems**

Future iterations can incorporate integration with Learning Management Systems (LMS), Enterprise Resource Planning (ERP) tools, and student information systems to automate data exchange and reduce manual effort.

### **Automated Accreditation Support**

The system can be extended to support real-time accreditation readiness by automatically generating reports aligned with NAAC, NBA, and ABET standards, thereby simplifying the compliance process.

### **Advanced Data Analytics and Dashboards**

Incorporating AI-driven analytics and interactive dashboards can provide deeper insights into student performance, course effectiveness, and outcome attainment, enabling data-informed decision-making.

### **Student and Parent Portals**

Introducing dedicated portals for students and parents can increase transparency, allowing them to track academic progress, outcome achievements, and course-related updates.

### **Mobile Application Support**

Developing a mobile version of the system would enhance accessibility and usability, allowing stakeholders to access key features and notifications on the go.

### **Automated Curriculum Recommendation Engine**

Leveraging AI to suggest curriculum improvements or personalized learning paths based on historical attainment data and industry trends can further optimize teaching strategies.

### **Training and Development Modules**

Adding modules for faculty training, OBE-related workshops, and certification tracking can ensure ongoing professional development and better system adoption.

By incorporating these advancements, the OBE Management System can evolve into a comprehensive academic management platform, capable of transforming educational quality, enhancing user experience, and ensuring sustainable institutional growth.

## Reference

- [1] <https://files.eric.ed.gov/fulltext/ED380910.pdf>
- [2] [https://saudijournals.com/media/articles/JAEP\\_52\\_58-65.pdf](https://saudijournals.com/media/articles/JAEP_52_58-65.pdf)
- [3] A HANDBOOK ON OUTCOME BASED EDUCATION (OBE) By: Team OBE Mr. (Ramgouda B. Patil, Mr. Mahesh A. Patil, Ms. S. V. Nikam) Department of Computer Science & Engineering (TKIET).
- [4] <https://phpgurukul.com/tourism-management-system-free-download/>
- [5] [https://youtu.be/dw1vLYg\\_ELo?si=ZA\\_a7ZQ2I93HhEYY](https://youtu.be/dw1vLYg_ELo?si=ZA_a7ZQ2I93HhEYY)