SOFTWARE REQUIREMENT SPECIFICATION

MARK ENTRY PLATFORM

Name	RAGHURAM C
Reg no	7376221CS272
Seat no	241
Project ID	14
Problem Statement	An interface for Five CO wise Periodical test marks, CO PO calculation, Result analysis

TECHNICAL COMPONENT:

Component	**MERN Stack**
Frontend	React(Js Framework)
Backend	Express.js(Web framework for Node.js) Node.js(Javascript runtime environment)
Database	MongoDB(NOSQL Database)
API	Open API

IMPLEMENTATION TIMELINE:

Phase	Deadline	Status	Notes
Stage 1	20/07/2024	Approved ⌄	Planning and requirement
Stage 2		In Progress ⌄	Design and Prototyping
Stage 3		Not Started ⌄	DB Designing
Stage 4		Not Started ⌄	Backend Implementation
Stage 5		Not Started ⌄	Testing & Implementation
Stage 6		Not Started ⌄	Deployment

1. INTRODUCTION :

1.1 Purpose :

The purpose of this document is to present a detailed description of the Mark Entry Portal. This system aims to streamline the process of recording, analyzing, and reporting student performance by calculating Course Outcome (CO) wise periodical test marks, CO-PO (Course Outcome - Programme Outcome) calculations, and result analysis. The portal is designed to enhance educational management and decision-making within our institution.

1.2. Scope of Project :

This project involves developing a comprehensive Mark Entry Portal with the following features:

- Secure login using the institution's email (BITSathy mail ID).
- CO-wise periodical test mark entry.
- CO-PO mapping and calculations.
- Detailed result analysis.
- Intuitive user interface.

2. SYSTEM OVERVIEW :

2.1 User Side :

- Secure Authentication: Secure login using bitsathy mail ID for staff.
- Department and Course Selection: Choose department, semester, and course ID (subject).
- Mark Entry: Enter marks for PT1, PT2, END SEMESTER, IP1, IP2, course exit survey, and articulation matrix.
- CO-PO Calculation: Calculate CO-PO attainment for academic insights.
- User Interface: Intuitive and accessible user interface.

2.2 Admin Side :

- Secure Authentication: Secure login for administrators.
- Course Management: Manage courses, COs, POs, and student data.
- Mark Management: Administrators can enter, edit, and organize test marks.
- Result Analysis: Comprehensive result analysis and reporting.
- Platform Configuration: Configure settings and manage permissions.

2.3. Features :

1. **User Authentication:** Secure login system for users and administrators.
2. **Mark Entry:** Ability for users to enter CO wise periodical test marks.
3. **CO-PO Calculation:** Calculation of CO-PO attainment based on entered marks.
4. **Result Analysis:** Detailed analysis of student performance and result trends.

3. FUNCTIONAL REQUIREMENTS :

1.User Authentication:

- Users and administrators can register and log in securely using BITsathy mail ID.
- Administrators should have additional privileges for managing courses, students, and marks.

2.Mark Entry:

- Users can enter CO-wise periodical test marks for PT1, PT2, END SEMESTER, IP1, IP2, course exit survey, and articulation matrix.
- Administrators can manage and edit entered marks.
- The system should specify details such as test name, date, and associated COs.
- For PT1 (CO1: 20 marks, CO2: 20 marks, CO3: 10 marks), PT2 (CO3: 10 marks, CO4: 20 marks, CO5: 20 marks), and END SEMESTER (5 COs each 20 marks), list the number of students and their names who scored less than 50% for each CO.
- For IP1 and IP2, each has a maximum mark of 8. IP1 can be split into CO1, CO2, CO3, and IP2 into CO3, CO4, CO5.

3.CO-PO Calculation:

- The system should provide a detailed analysis of student performance based on entered marks.
- Administrators can generate reports on overall student performance, CO attainment, and PO attainment.
- Course exit survey ratings out of 5 will be converted into a rating out of 3 for each CO.
- The articulation matrix will have 5 COs, 12 POs, and 3 PSOs attainment, each rated out of 3 (minimum 1 to maximum 3, no decimal values).

4.NON-FUNCTIONAL REQUIREMENTS :

1.Performance:
- The website should respond quickly to user interactions, with minimal latency.
- It should support a large number of concurrent users without significant performance degradation.

2.Reliability:

- The website should be available and operational at all times, with minimal downtime for maintenance or updates.
- It should have mechanisms in place to handle and recover from system failures gracefully.

3.Usability:

- Users should need minimal instruction to operate the user interface since it should be simple to use and intuitive.
- It should adhere to accessibility standards to accommodate users with disabilities.

4.Security:

- Encryption techniques should be used to securely store and transfer user data.
- The website should implement strong authentication mechanisms to prevent unauthorized access.

5.Scalability:
- The website should be able to scale horizontally to handle increasing load by adding more servers or resources.

5. FLOWCHART :