SOFTWARE REQUIREMENT SPECIFICATION

EVENT MANAGEMENT

Name	SOPHIGA R
Roll no	7376221BM145
Seat no	**168**
Project id	**30**
Problem Statement	CURRICULUM AUTOMATION

TECHNICAL COMPONENT:

Component	**Tech Stack**
Frontend	● HTML ● CSS ● JS ● ANGULAR
Backend	● Python
Database	● Mongo DB
API	● Node.js ● Express.js

IMPLEMENTATION TIMELINE:

Phase	Deadline	Status	Notes
Stage1	**25/07/2024**	IN PROGRESS	Planning and requirement
Stage2		IN PROGRESS	Design and Prototyping
Stage3		Not Started	DB Designing
Stage4		Not Started	Backend Implementation
Stage5		Not Started	Testing &Implementation
Stage6		Not Started	Deployment

1. INTRODUCTION:

Purpose

The purpose of the curriculum automation project is to streamline course management processes in educational institutions by automating course code allocation, LTPC verification, duplicate entry prevention, exam pattern assignment, and syllabus entry. This enhances administrative efficiency, ensures data accuracy, and simplifies the overall curriculum management workflow.

Scope of the Project

The project scope covers creating a web application to manage course data, ensuring real-time validation, and preventing duplicate entries. It integrates course code allocation, LTPC verification, exam pattern assignment, and syllabus entry, aiming to improve administrative efficiency and data accuracy in educational institutions

2.Overall Description

Product Perspective

The product automates curriculum management, improving efficiency, accuracy, and ease of use in educational institutions. Product Functions

- **Efficiency:** Automates manual curriculum management processes, reducing administrative workload and saving time.
- **Accuracy:** Ensures precise data management, minimizing human errors in course code allocation and LTPC verification.
- **Consistency:** Maintains uniformity across all curriculum-related tasks, preventing discrepancies and duplicate entries.
- **Scalability:** Adapts to the needs of educational institutions of varying sizes, supporting growth and changes.
- **User-Friendly:** Offers an intuitive interface for administrators, simplifying curriculum management and enhancing user experience.

User Classes and Characteristics

- Administrators and faculty members with basic computer skills will use the system for managing course data efficiently.
- Includes Course, LTPC, User, and Exam Pattern classes to structure and handle the curriculum data effectively.

Operating Environment

The operating environment consists of a web server running Node.js and Express, with MongoDB for database management. Users access the application through modern web browsers on desktop or mobile devices.

Design and Implementation Constraints

- **Browser Compatibility**: The application must support all major web browsers to ensure accessibility for all users.

- **Data Security**: Implementation must include robust security measures to protect sensitive curriculum data from unauthorized access.

- **Performance:** The system must handle high volumes of data and concurrent users efficiently without significant performance degradation**.**

User Documentation

- Quick Start Guide
- User Manual
- FAQ and Troubleshooting

Assumption sand Dependencies

- Stable Internet Connection:
- Database Availability:

2. System Features

Event Management

Description and Priority

Indicates the urgency and importance of implementing each feature or component.

Functional Requirements

- Validates lecture, tutorial, practical, and credit details in real-time to ensure data accuracy.
- Automatically detects and prevents duplicate course codes and data entrie

3. External Interface Requirements

User Interfaces

The user interface must be intuitive and user-friendly, providing easy navigation for administrators and faculty members. It should include clear, accessible menus and forms for managing course details, LTPC verification, and exam pattern assignments. Responsive design is essential to ensure compatibility across various devices and screen sizes.

Hardware Interfaces

- Server
- workstation,
- network
- storage
- backup

Software Interfaces

- Database
- web browser
- middleware
- Authentication.

Communications Interfaces

- HTTP
- HTTPS
- REST ful API
- Web Socket
- JSON.

4. Other Non-Functional Requirements

Performance Requirements

- System must handle concurrent users efficiently with minimal response time.

Safety Requirements

- Safety requirements ensure the system protects data integrity, prevents unauthorized access, and provides regular backups to avoid data loss.

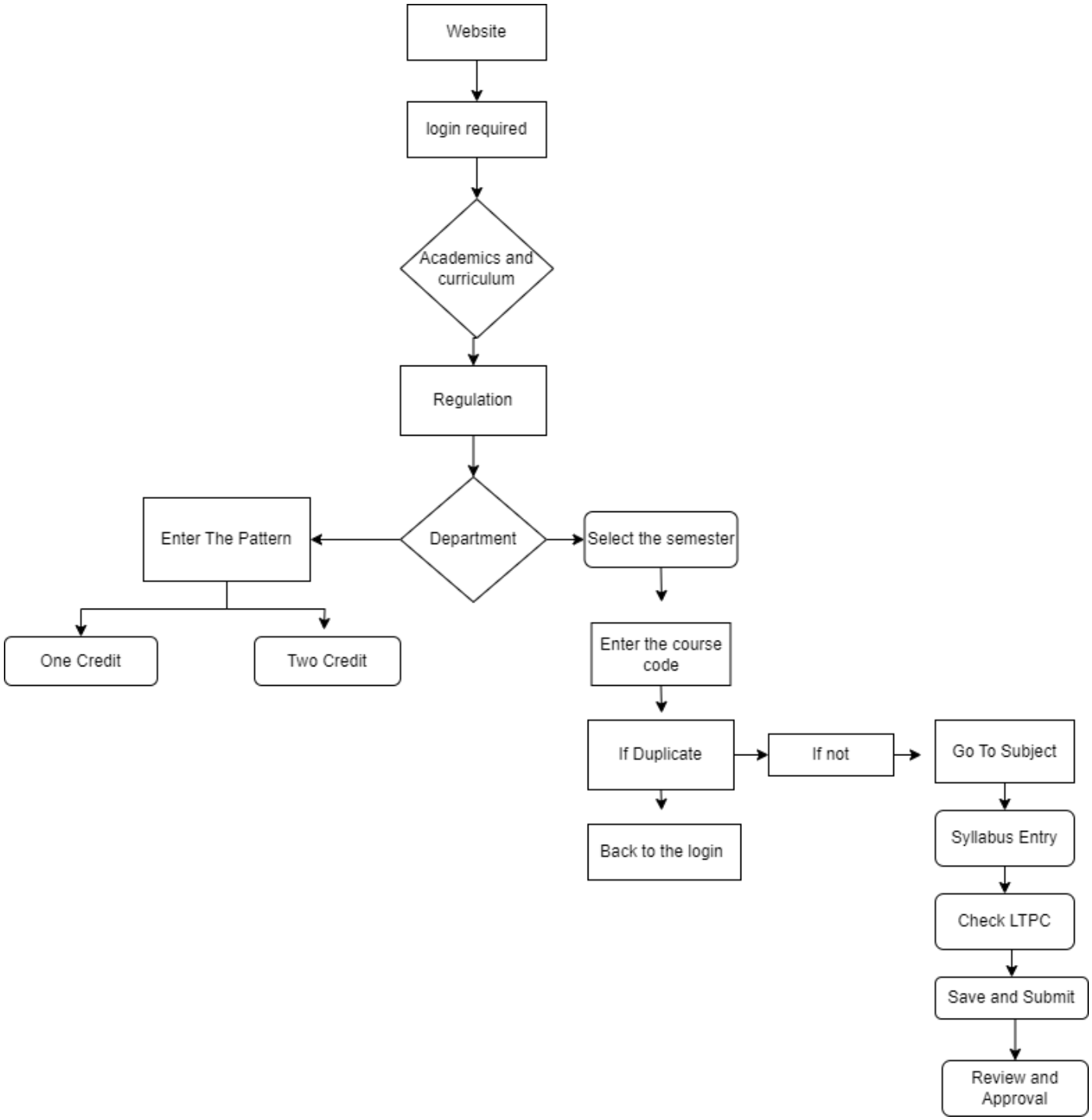
Security Requirements

- Security requirements include user authentication and authorization, data encryption, secure API access, regular security audits, and protection against common threats like SQL injection and cross-site scripting (XSS).

Software Quality Attributes

- Reliability,
- Performance
- Usability
- Maintainability,
- Scalability.

5. FLOWCHART


Website
login required
Academics and curriculum
Regulation
Department
Enter The Pattern
One Credit
Two Credit
Select the semester
Enter the course code
If Duplicate
Back to the login
If not
Go To Subject
Syllabus Entry
Check LTPC
Save and Submit
Review and Approval