SOFTWARE REQUIREMENT SPECIFICATION FOR CURRICULUM AUTOMATION

NAME : SHRIDHAR R

ROLL NO : 7376222AL201

PROJECT ID : 30

PROJECT NAME : CURRICULUM AUTOMATION

TASK

Course code allocation, checking LTPC, avoiding duplicates, exam pattern assignment and syllabus entry

TECHNOLOGY STACK:

Front end	HTML,CSS,JS
Backend	PYTHON DJANGO
Database	MYSQL
API	Restful API

PROJECT OVERVIEW:

The Course Management System project aims to streamline the management of courses within an educational institution. The system will handle the allocation of course codes, validation of LTPC (Lecture-Tutorial-Practical-Credit) values, avoidance of duplicate course entries, assignment of exam patterns, and entry and management of syllabi.

KEY OBJECTIVES:

1. **Automate Course Code Allocation**: Ensure unique and systematic course code generation.
2. **Validate LTPC Values**: Maintain accurate and institutionally compliant LTPC values.
3. **Prevent Duplicate Entries**: Implement mechanisms to avoid duplication of course codes and information.
4. **Standardize Exam Patterns**: Consistently assign and manage exam patterns across courses.
5. **Efficient Syllabus Management**: Facilitate the entry, updating, and retrieval of comprehensive course syllabi.

FUNCTIONAL REQUIREMENTS:

1. Course Code Allocation

- Generate unique course codes based on department, level, and sequence.
- Allow manual entry of course codes with duplication checks.

2. LTPC Validation

- Ensure LTPC values are in the correct format (L-T-P-C).
- Calculate total credits based on L, T, and P values.
- Check that L, T, P, and C values are within acceptable ranges.

3. Duplicate Prevention

- Implement checks to avoid duplicate course codes during data entry.
- Maintain a log of course code assignments and modifications.

4. Exam Pattern Assignment

- Define different types of exams (e.g., Midterm, Final).
- Assign weightage to each exam type.
- Manage exam schedules.
- Specify assessment methods (e.g., multiple-choice, essays).

5. Syllabus Management

- Enter and store course descriptions.
- Specify learning outcomes.
- Provide a weekly course outline.
- List recommended reading materials.
- Outline assignments and projects.
- Detail assessment criteria.

NON-FUNCTIONAL REQUIREMENTS:

1. **Performance**

- The system should handle multiple concurrent users without performance degradation.
- Quick response time for data entry, validation, and retrieval operations.

2. **Scalability**

- The system should accommodate growth in the number of courses and users.
- Easily adaptable to include additional features in the future.

3. **Security**

- Ensure data security and privacy through authentication and authorization mechanisms.
- Implement data encryption and secure access protocols.

4. **Usability**

- Provide a user-friendly interface for easy navigation and data entry.
- Offer comprehensive help and support features for users.

5. **Reliability**

- Ensure high availability and minimal downtime.
- Implement regular data backup and recovery processes.

PROJECT DELIVERABLES:

1. **System Design Documentation**

    - Detailed design specifications and architectural diagrams.
    - Database schema design and data flow diagrams.

2. **Implementation**

    - Development of the Course Management System application.
    - Integration with existing institutional systems, if applicable.

3. **Testing**

    - Comprehensive testing plan including unit, integration, system, and user acceptance testing.
    - Bug tracking and resolution documentation.

4. **User Training and Documentation**

    - User manuals and training materials.
    - Training sessions for end-users and administrators.

5. **Deployment and Maintenance**

    - Deployment plan and schedule.
    - Ongoing maintenance and support plan.

WORKFLOW: