

# Self Intensive Training - FullStack Development

[Software Requirement Specification Document ]

**Student Name:** PRIYADHARSHINI S

**Roll no:**7376221SE139

**Seat No:** 171      **Project ID:** 30

**Project Title:**Curriculum Automation

## Technical Components :

Component Tech Stack(LAMP)
<b>Linux:</b> The operating system hosting the server environment.
<b>Apache:</b> The web server that handles HTTP requests and serves web content
<b>MySQL:</b> The relational database system storing curriculum data like courses,syllabi and exam patterns.
<b>PHP:</b> The server-side scripting language for backend logic, interactive with the database, and generating dynamic web pages.

## Workflow Timeline :

Phase	Deadline	Status Stages
Stage 1	24/07/2024	In Review Planning & Requirement Gathering
Stage 2		On Progress Design & Prototyping
Stage 3		On Progress Backend Development
Stage 4		On Progress DB Design & Implementation
Stage 5		Not Started Integration & Testing
Stage 6		Not Started Deployment

## **PROBLEM STATEMENT:**

**Project Title :** Curriculum Automation

**Module Description :** Course code allocation, checking LTPC, avoiding duplicates, exam pattern assignment and syllabus entry.

Course code allocation involves assigning unique identifiers to each course to ensure easy tracking and management. Checking LTPC (Lecture, Tutorial, Practical, Credit) ensures that courses have the correct distribution of hours and credits. Avoiding duplicates is crucial to prevent multiple courses from having the same code, which can lead to confusion and errors. Exam pattern assignment entails defining the structure and format of assessments, including types of exams and grading criteria. Syllabus entry involves inputting detailed curriculum information, course objectives, learning outcomes, and other relevant details into the system for reference and planning purposes.

## **FEATURES:**

### **Course Code Allocation:**

Course code allocation assigns unique identifiers to each course to ensure clear identification, efficient tracking, and management within an academic system. This process prevents confusion and errors by ensuring no two courses share the same code.

### **Checking LTPC:**

Checking LTPC involves verifying the distribution of Lecture, Tutorial, Practical hours, and Credits for each course. This ensures that the course meets academic requirements and standards, providing a balanced and comprehensive learning experience for students.

### **Avoiding Duplicates:**

Course code allocation assigns unique identifiers to courses for easy tracking. Checking LTPC ensures the correct distribution of Lecture, Tutorial, Practical, and Credit hours. Avoiding duplicates prevents multiple courses from having the same code, avoiding confusion. Exam pattern assignment defines the structure of assessments for each course. Syllabus entry involves inputting detailed curriculum, course objectives, and learning outcomes into the system for comprehensive academic planning.

### **Exam Pattern Assignment:**

Exam pattern assignment defines the structure and format of assessments for each course. It involves setting types of exams, grading criteria, and weightage of each assessment component. This process ensures consistency and fairness in evaluations, aligns with course objectives, and integrates with the academic calendar for scheduling, providing a clear framework for both instructors and students.

### **Syllabus Entry:**

Syllabus entry involves inputting detailed course information into the academic system. This includes the curriculum content, learning objectives, course materials, and weekly topics. It ensures that all necessary details are available for students and instructors, facilitates easy updates, and provides a centralized database for consistent access. This process supports academic planning, transparency, and alignment with institutional standards and learning outcomes.

### **IMPLEMENTATION STRATEGY :**

1. Requirement Analysis: Gather and document detailed requirements from stakeholders and analyze current processes to identify needs for course code allocation, LTPC verification, duplicate avoidance, exam patterns, and syllabus entry.
2. System Design: Develop system architecture, user interfaces, and database schema. Create flowcharts and wireframes for user interactions.
3. Development: Build features for code allocation, LTPC checking, duplicate detection, exam pattern management, and syllabus entry.
4. Testing: Perform unit, integration, and user acceptance testing to ensure functionality.
5. Deployment: Deploy the system, migrate data, and train users.
6. Maintenance and Support: Monitor system performance, address issues, and provide updates.
7. Documentation: Prepare user manuals and technical guides.

## **USE CASES**

- An academic administrator inputs a new course into the system, which automatically generates a unique code, ensuring no duplication and easy course identification.
- A faculty member submits a course for approval; the system checks LTPC values against predefined standards and alerts if there are discrepancies, ensuring compliance with academic requirements.
- When entering a new course, the system checks for existing courses with the same code or title and provides alerts if duplicates are found, maintaining data integrity.
- An instructor sets up an exam pattern for a course, defining types of assessments, weightage, and grading rubrics, which are then integrated into the course management system.
- A course coordinator enters the syllabus details into the system, including course content and objectives, which are then stored in a centralized database for easy access and updates.

## **LTPC CALCULATION:**

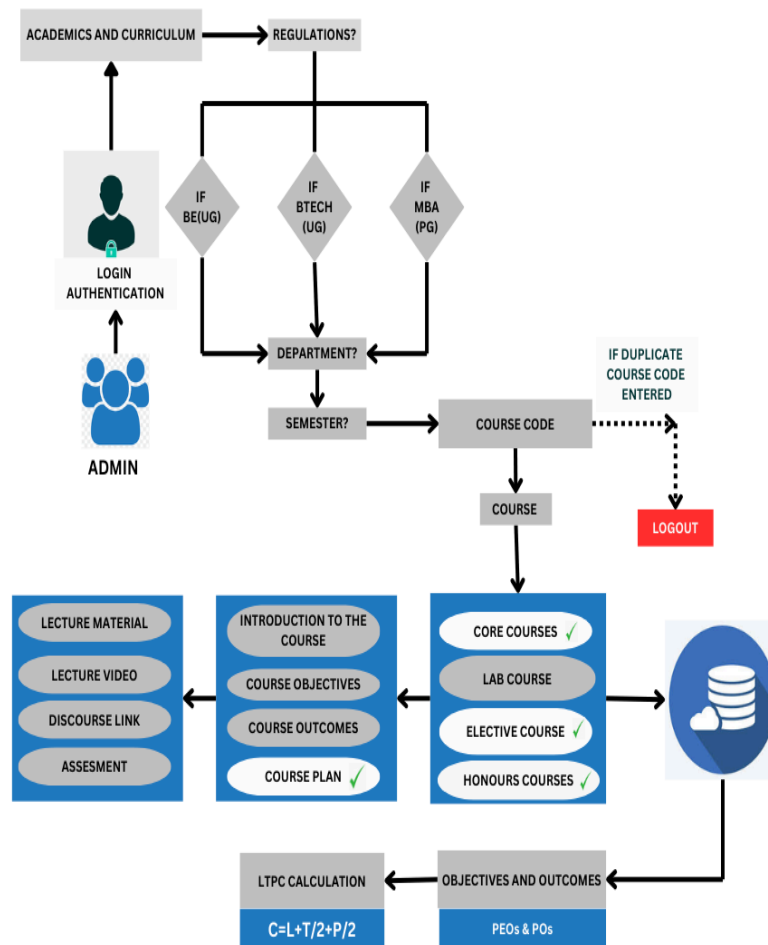
LTPC stands for Lecture (L), Tutorial (T), Practical (P), and Credit (C), and it's a system used in many academic institutions to represent the workload and credit value of a course. The calculation of LTPC generally follows these guidelines:

1. Lecture (L): This represents the number of lecture hours per week. Typically, each hour of lecture per week over a semester is equivalent to one credit.
2. Tutorial (T): This indicates the number of tutorial hours per week. Tutorials often involve small group discussions, problem-solving sessions, or other interactive learning activities.
3. Practical (P): This stands for the number of practical or lab hours per week. Practical hours involve hands-on learning, laboratory work, or practical exercises.

4. Credit (C): Credits are usually a measure of the total workload of a course, combining lectures, tutorials, practicals, and self-study. The credit calculation can vary by institution, but a common formula is:

$$\text{TOTAL CREDITS}(C)=L+T/2+P/2$$

### FLOWCHART OF ADMIN CURRICULUM AUTOMATION:



## FLOWCHART OF STUDENT CURRICULUM AUTOMATION:

