

PROJECT FLOW

Name	VIJAY S R
Roll no	7376222AD222
Seat no	140
Project ID	20
Problem Statement	Online Course System

1. Introduction

1.1. Problem Statement:

In today's fast-paced digital world, the need for adaptable, dynamic, and engaging learning platforms has never been greater. This project intends to transform online education by creating a comprehensive Online Course System. Our platform will give students easy access to a wide range of courses, allow them to connect with qualified instructors in real time, and measure their progress individually. Administrators and instructors will have powerful tools for effectively managing courses, evaluating student performance, and fostering a collaborative learning environment. This system will bridge the gap between conventional learning and current technology, providing a dynamic and scalable solution for digital-era education.

1.2. Scope of Project:

- The Online Course System will provide a comprehensive platform for users to browse, search, enroll, and participate in online classes. It will make course design and maintenance easier for instructors, allow users to post reviews and ratings, visualize course outputs, track progress, submit tasks (including video uploads), and communicate with instructors via a chat board.
- To support a huge user base and course catalog, the system will prioritize responsiveness, usability, and scalability.

2. System Overview:

2.1. Users:

1. Admin:

They have the ability to Create, update, and delete courses. And they can view completed tasks by students.

2. Instructor:

Evaluate student tasks, provide remarks, and communicate with students via chat board.

3. User (Student):

Enroll in courses, learn, submit tasks, and interact with instructors.

2.2. Features:

1. User Authentication:

Secure login and registration for users, admins, and instructors.

2. Course Management:

- Admin: Create, update, and delete courses.
- Instructor: Evaluate student tasks and communicate with students.

3. User Interaction:

Students can post questions on the chat board, and instructors can answer.

4. Progress Tracking:

Keep track of the user's progress; if they are inactive for more than 48 hours, ask them to repeat the course.

5. Task Submission:

Students submit source files, output screenshots, and videos for evaluation by instructors.

6. Approvals and Notifications:

The administrator approves students based on their grades and sends notifications to the users.

7. Leaderboard:

Display top-performing students based on course-wise marks.

3. System Requirements Specification:

3.1 Functional Requirements:

● Admin Functions:

- Access: Full administrative access.

- Create, update, and delete courses.
- View completed tasks by students.
- Approve students based on marks for course completion.

- **Instructor Functions:**

- Access: Evaluate student tasks and communicate with students.
- Responsibilities:
 - Evaluate tasks submitted by students.
 - Communicate with students via chat board.

- **User (Student):**

- Access: Enroll in courses and access learning materials.
- Responsibilities:
 - Enroll in courses.
 - Learn and complete tasks.
 - Submit questions on the chat board.
 - Submit tasks after course completion.
 - Track performance on the leaderboard.

- **Progress Tracking:**

- Detect user inactivity; prompt restart if inactive for more than 48 hours.

- **Admin Dashboard:**

- Admins can view a list of all students who submitted the tasks.
- Admins can approve or reject student with suitable remarks

- **Approvals and Notifications:**

- Admin approves students based on marks.
- Notifications sent to users for course completion status.

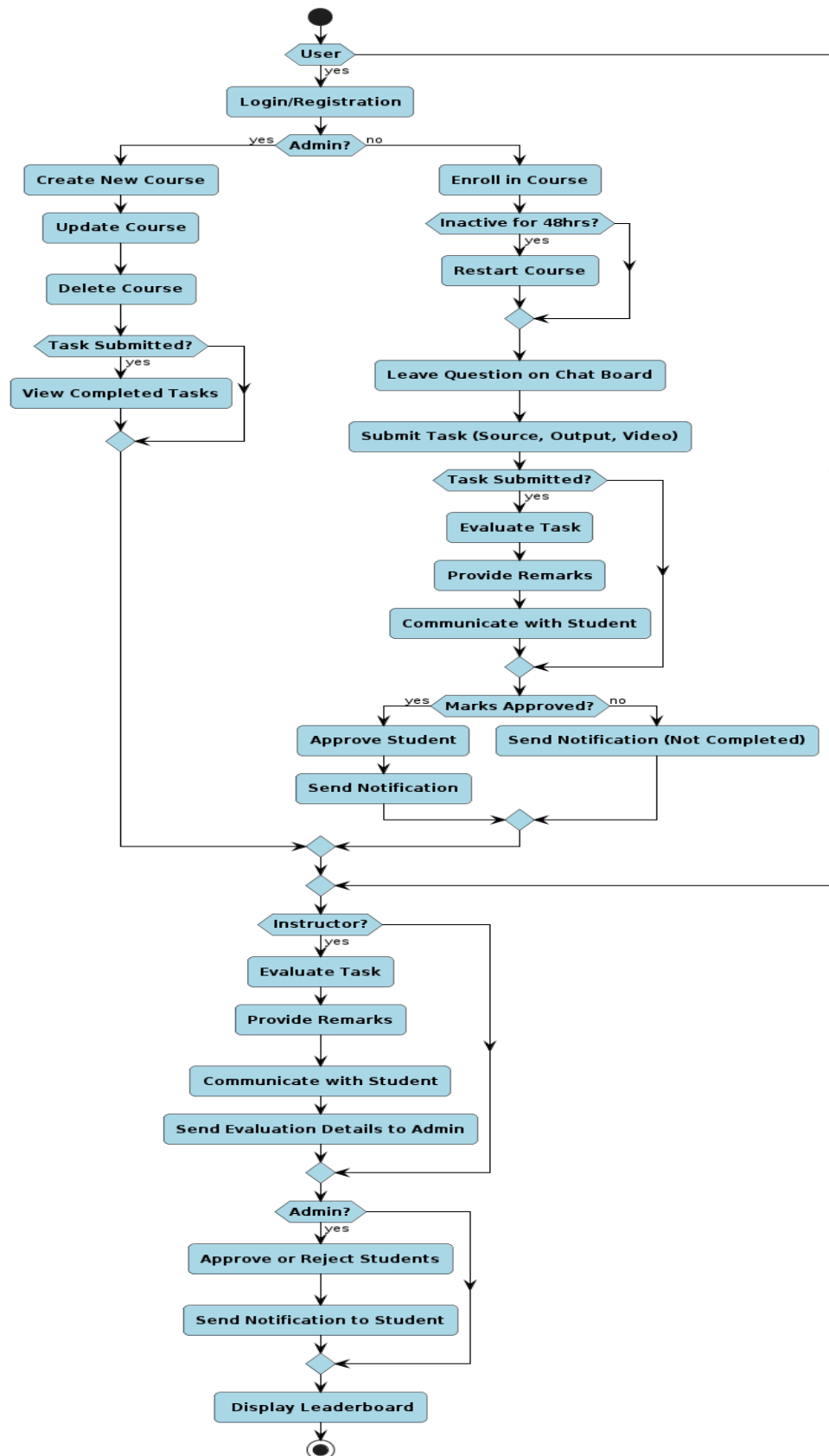
- **Leaderboard Dashboard:**

- Display top-performing students based on course-wise marks.

3.2. Non-Functional Requirements:

- **Performance:** The system must respond to user actions within 2 seconds to ensure efficient usability and must handle a concurrent user load of at least 100 users without significant performance degradation.
- **Security:** User data must be encrypted during transmission and storage, and access to sensitive functionalities should be restricted to authorized admin users through secure authentication mechanisms.
- **Usability:** The user interface should be intuitive and user-friendly, with clear and concise error messages provided to guide users in case of input errors or system failures.
- **Reliability:** The system should be available 24/7 with minimal downtime and should have a backup and recovery mechanism in place to prevent data loss in case of system failures or crashes.
- **Scalability:** The system should be designed to accommodate an increasing number of users and data volume over time, and it should be scalable to support additional features and functionalities as per future requirements.

4. Flowchart:



5. Technical Components:

Component	Tech Stack
Backend	Express and Node JS
Frontend	React JS
Database	Mongo DB
API	Restful services