Software Requirement Specification for Online Course System

Name	Tharun Teja G
Roll no	7376222IT271
Seat no	340
Project ID	20
Problem Statement	Online Course System

1. Introduction

This document outlines the Software Requirements Specification (SRS) for an online course system. The system is also equipped with administrative functionalities for course creation, instructor assignment, and monitoring learner progress. This system aims to provide a comprehensive and interactive learning environment for learners, instructors, and administrators.

1.1 Purpose:

The Online Course System is designed to provide a comprehensive and interactive platform for learners to enroll in courses, watch course materials, complete weekly tasks, engage in private chat with instructors, submit tasks in various formats, receive feedback, and track their progress. It aims to enhance the learning experience by ensuring engagement with course materials and timely completion of tasks.

1.2 Scope of Project:

- The main scope of this project is to develop a web-based online course system that facilitates a **structured and interactive learning environment** for three user roles (Admin, Instructor, and learner). Submitted tasks are reviewed and evaluated by instructors.

- And the core functionalities are course creation and delivery, learner progress and interaction, and admin oversight. This project aims to build a system that supports effective online learning by providing a structured framework for course delivery, learner progress tracking, and instructor-learner interaction. From an administrative perspective, this online course system will empower you with a powerful and comprehensive analytical dashboard for comprehensive project oversight.

2.System Overview:

2.1 Users:

1. Admins:

Admin Creates courses with video lectures and weekly tasks, sets time limits for course completion, setting titles, descriptions, and assigns instructors. They have the ability to view learner's progress and assign one or more instructors to each course. Admins should set a range of time and if that limit exceeds the learner should view the lectures from the beginning. And the tasks should be more than one and each and every learner should be assigned with random tasks.

2. **Instructors:**

The primary responsibility of the instructor is to clarify the doubts of the learners via private chat and to provide additional explanations, offer support throughout the course duration. Instructors review and provide feedback on learner submissions, offering constructive comments to help learners improve their understanding and skills. Learners can clarify doubts through private chat with instructors. Instructors monitor learner progress, keeping track of course completion status, task submissions, and overall performance. They use this information to provide personalized guidance and support to learners as needed.

3. **Learners:**

Learners can register for an account. Users can log in using their credentials. Users can update their profiles. Learners can discover and enroll in various courses offered by various admins. Learners can progress through the coursework by watching video lectures, completing weekly tasks in different formats (code, images, videos, graphics design), and submitting them for instructor evaluation. Learners can monitor their progress through a personalized progress tracker and leaderboard (which resets upon every user progress).

2.2 Features:

1. **User Management:**

Students can register for an account or login with their existing account. Users can register for an account. Registered users can log in using their credentials. Users can update their profiles with personal information.

2. **Course Management:**

   Learners can browse available courses and enroll in them. Admins can set time limits for course completion. Admins can assign one or more instructors to each course.And most importantly the videos should not be fast forwarded and completed. That should be watched fully to get completion status.

3. **Learning Activities:**

   Learners must watch course videos fully to mark them as complete. Instructors assign weekly tasks based on course content. Learners submit tasks in various formats such as code, images, videos, and graphics design. Instructors verify submitted tasks and provide remarks.

4. **Communication:**

   Learners can engage in private chat with instructors to clarify doubts and discuss course content.

5. **Progress Tracking:**

   The system tracks learner progress, including course completion status and task submissions. A leaderboard displays the progress of learners and resets with every update.

6. **Interface Design:**

   Provides an intuitive interface for learners to navigate courses, submit tasks, and engage with instructors. Allows admins to create courses, assign instructors, and monitor learner progress. Enables instructors to assign tasks, verify submissions, and communicate with learners.

7. **Additional Features:**

   Learners can proceed to the next course or next assessments only if the particular tasks are marked complete. Otherwise, the lectures should be watched again to get completion status. Once a learner gets completion status, they should be assigned with tasks. If a learner struggles with a task or concept, they can rewatch specific video sections.

8. **Additional Features:**

   - **Content Security:** Protects course materials from unauthorized access.
   - **Scalability:** The system can accommodate a growing number of users and courses.
   - **Accessibility:** The system should be accessible to users with disabilities.

3. System Requirements Specification:

3.1 Functional Requirements:

- **Registration:**
  - Users should be able to register for an account by providing necessary information such as name, email address, and password.
  - The system should validate user input to ensure data integrity and security.

- **Login:**
    - Registered users should be able to log in using their email address and password.
    - The system should authenticate user credentials and grant access to the appropriate features based on user roles (learner, instructor, admin).

- **Profile Management:**
    - Users should be able to update their profile information, including name, email address, profile picture, and password.
    - The system should allow users to view and edit their profile details easily.

- **Course Creation:**
    - Admins should have the ability to create new courses by providing course details such as title, description, duration, and prerequisite knowledge.
    - The system should support the addition of multimedia course materials such as videos, documents, and presentations.

- **Course Enrollment:**
    - Learners should be able to browse available courses and enroll in them.
    - The system should display course details and availability status to help learners make informed enrollment decisions.

- **Instructor Assignment:**
    - Admins should be able to assign one or more instructors to each course.
    - The system should allow admins to select instructors based on their expertise and availability.

- **Watch Course Materials**
    - Learners should be required to watch course videos fully to mark them as complete.
    - The system should track video progress and update completion status accordingly.

- **Weekly Tasks:**
    - Instructors should assign weekly tasks to learners based on course content and learning objectives.
    - The system should display task descriptions, deadlines, and submission requirements to learners.

- **Task Verification:**
    - Instructors should review submitted tasks, verify their accuracy and completeness, and provide feedback to learners.
    - The system should notify learners of task verification status and display instructor remarks

- **Private Chat:**
    - Learners should be able to engage in private chat with instructors to ask questions, seek clarification, and discuss course content.
    - The system should provide a secure and user-friendly chat interface with features such as message history and real-time notifications.

- **Progress Tracking:**
    - The system should track learner progress throughout the course, including course completion status, task submissions, and grades.
    - Learners should be able to view their progress and performance metrics in a centralized dashboard.

- **Leaderboard:**
    - The system should maintain a leaderboard that displays the progress of learners, ranking them based on their performance and activity.
    - The leaderboard should update dynamically based on learner achievements and reset periodically to reflect current standings.

- **Graphical User Interface:**
    - Designing a modern and intuitive course interface with interactive elements like buttons, menus, and sliders for seamless browsing.

    - Integrating multimedia elements such as videos, slideshows, and interactive quizzes within the course interface to enrich the learning experience.

- Providing learners with a dedicated task submission interface that supports drag-and-drop functionality and provides immediate feedback on submissions.

3.2. Non-Functional Requirements:

Performance:

- **Responsiveness:**
  - The system should display pages and respond to user actions within a specified timeframe (e.g., page load time under 3 seconds).
  - This ensures a smooth and uninterrupted user experience.

- **Scalability:**
  - The system should be able to accommodate a growing number of users, courses, and content without significant performance degradation.
  - This ensures the platform can handle future growth effectively.

- **Availability:**
  - The system should be available for access by authorized users with minimal downtime (e.g., 99.9% uptime).
  - This minimizes disruption to learning activities.

Security:

- **User Authentication and Authorization:**
    - The system should implement secure user authentication mechanisms (e.g., username/password login with strong password requirements) to prevent unauthorized access.
    - Role-based access control should restrict functionalities based on user roles (admin, instructor, learner).
- **Data Security:**
    - Learner data, course content, and system configurations should be protected from unauthorized access, modification, or disclosure.
    - Encryption of sensitive data and secure storage practices are crucial.

Usability:

- **Intuitive Interface:**
    - The user interface should be clear, consistent, and easy to navigate for all user roles.
    - Minimal cognitive load should be required to complete tasks.

- **Accessibility:**
    - The system should be designed to be accessible to users with disabilities, adhering to relevant accessibility standards (e.g., WCAG 2.1).
    - This ensures equal access to the learning platform for all users.

Maintainability:

- **Modular Design:**
  - The system should be designed with modular components to facilitate future modifications and updates.
  - This simplifies maintenance and reduces development costs.

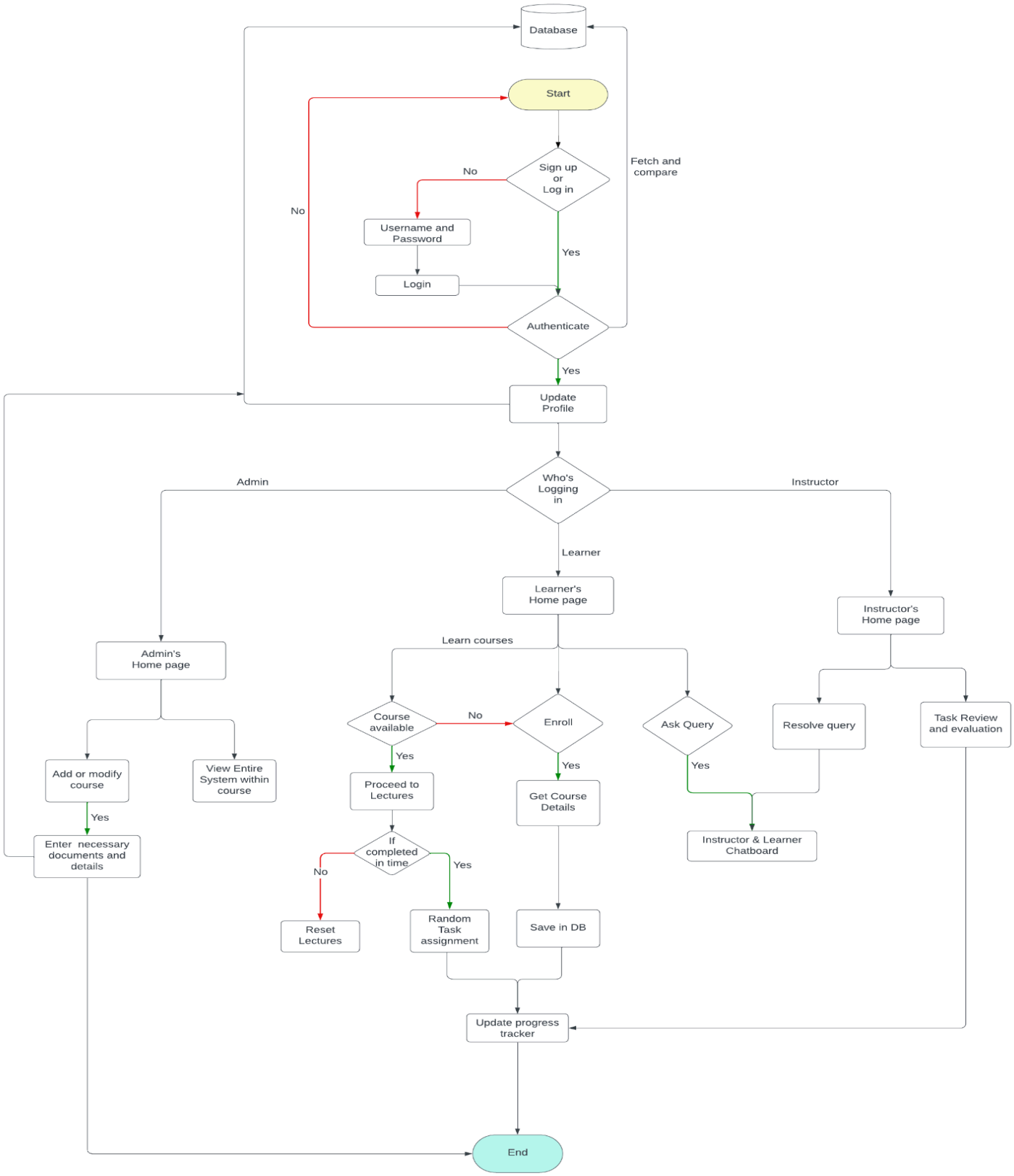
- **Code Documentation:**
  - The system code should be well-documented with clear explanations and comments.
  - This aids future developers in understanding and maintaining the codebase.

Additional Considerations:

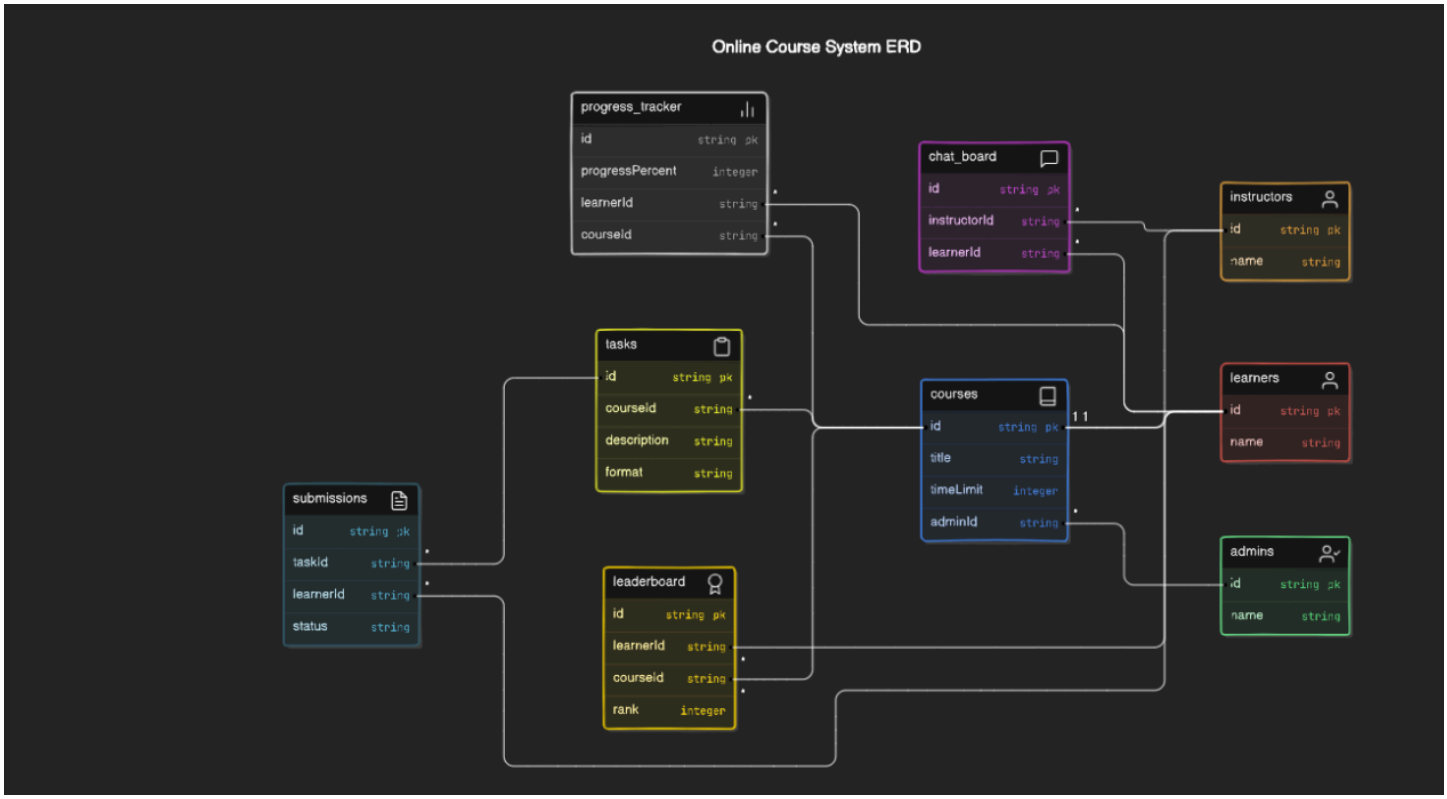
- **Logging and Monitoring:**
  - The system should implement logging mechanisms to track user activity, system events, and potential errors.
  - This facilitates troubleshooting and performance analysis.

- **Error Handling:**
  - The system should handle errors gracefully, providing informative messages to users and logging details for further investigation.
  - This ensures a smooth user experience even when unexpected issues arise.

FLOWCHART


Database
Start
Sign up or Log in
No
Fetch and compare
No
Username and Password
Yes
Login
Authenticate
Yes
Update Profile
Who's Logging in
Admin
Instructor
Learner
Learner's Home page
Instructor's Home page
Admin's Home page
Learn courses
Course available
No
Enroll
Ask Query
Resolve query
Task Review and evaluation
Add or modify course
View Entire System within course
Yes
Proceed to Lectures
Yes
Get Course Details
Yes
Yes
Enter necessary documents and details
If completed in time
No
Yes
Instructor & Learner Chatboard
Reset Lectures
Random Task assignment
Save in DB
Update progress tracker
End

ER (Entity Relationship) Diagram:



Stack:

Front End	HTML, CSS, JS
Backend	Python, Django (Python Web)
Data Base	MySQL, PostgreSQL
API	Open API, SOAP APIs, RESTFul API