Online course system

**TEACHER ROLE:**

**Teachers** are the core content creators and facilitators in the system. They are responsible for creating, managing, and overseeing the progress of students within the course.

1. **Upload Course Materials:**
   * Teachers can upload course materials such as **lessons**, **videos**, **presentations**, **documents**, etc.
   * These materials are made accessible to students for their learning.
   * For example, a teacher may upload a **Java programming module** that includes lecture notes and a video.
2. **Manage Assignments:**
   * Teachers can create and **assign tasks or assignments** for students based on the course content.
   * They set deadlines for assignments and keep track of submissions.
   * For instance, after completing a lesson on "Introduction to Python," the teacher assigns a coding task to be completed and submitted by students.
3. **Oversee Student Progress:**
   * Teachers monitor student performance through **submissions** and **test results**.
   * They can review the **quality** and **timeliness** of submitted assignments, providing feedback and grades accordingly.
   * Teachers may set **reminders** or **notes** on assignments to guide students in their learning journey.
4. **Issue Certificates:**
   * After students complete the course and fulfill the requirements (like submitting assignments and passing tests), teachers can issue **certificates**.
   * These certificates serve as proof of completion and may be used by students for their career development.
   * Teachers can customize certificates with course name, student details, and completion date.

**STUDENT ROLE:**

**Students** are the main users who interact with the system to **register** for courses, **submit assignments**, **check their grades**, and receive **certificates**.

1. **Register:**
   * Students must first **create an account** to join the platform.
   * They can register by providing **personal details** like name, email, etc.
   * Once registered, they can explore available courses and choose what they want to study.
2. **Search and Access Courses:**
   * Students can **browse** and **search** for courses offered by various teachers.
   * Once they find a course of interest, they can **enroll** in it and start learning.
   * For example, a student may search for courses on **“Data Science”** or **“Web Development”**.
3. **Submit Assignments:**
   * After completing lessons, students are required to **submit assignments** for evaluation.
   * They can upload their **work** (documents, code, reports, etc.) directly through the platform.
   * Submitting assignments on time is essential to progress in the course and receive grades.
4. **View Grades:**
   * Once the teacher grades the assignments or tests, students can **check their performance** in the course.
   * Students can view **individual grades** for each assignment or a summary of their overall score.
   * This feature helps students track their progress and know where they need to improve.
5. **View Certificates:**
   * After successfully completing the course, students can **download or view certificates** issued by the teacher or platform.
   * These certificates are essential for showcasing the student’s achievements to future employers or educational institutions.

**ADMIN ROLE:**

**Admins** are the system's **supervisors**. They manage the user accounts, course content, and overall platform functionality.

1. **Manage Users (Teachers and Students):**
   * Admins can **add, update, or delete user profiles** (both teachers and students).
   * Admins also manage roles—assigning permissions to users to ensure they have access to appropriate parts of the system.
   * For example, if a teacher leaves the platform, the admin can **remove their account** or update their profile information.
2. **Manage Course Content:**
   * Admins ensure that all courses and educational content are up-to-date and available.
   * They may approve or reject courses that teachers upload, ensuring only relevant and high-quality materials are shared.
   * Admins also organize course categories and structure the course offerings for easier navigation.

**Functional Requirements:**

1. **Authentication:**
   * The LMS must provide a secure authentication system to confirm user identities using login credentials.
   * Users should enter their username and password to access their account.
   * Additional security measures such as **multi-factor authentication (MFA)** (e.g., SMS code or email verification) should be included to enhance security and prevent unauthorized access.
   * Roles such as **Admin**, **Teacher**, and **Student** must be authenticated to access the system's respective features and permissions.
2. **Profile Management:**
   * The system should allow users to **create and update** their personal information, such as name, email, phone number, and profile picture.
   * Users should be able to change settings related to notifications, privacy, and preferences.
   * Teachers and Admins should have access to additional profile management features, such as viewing and updating course-related settings or user roles.
3. **Course Management:**
   * Teachers (or Admins) should be able to **create, organize, and manage courses** within the LMS platform.
   * Course management features include uploading course materials (e.g., documents, videos), setting up modules or lessons, creating assignments, and managing course schedules.
   * Instructors should be able to **assign students** to specific courses and **track their progress** throughout the course.
   * Admins should have the ability to approve or modify courses, monitor student enrollments, and manage course categories.
4. **Assessment Management:**
   * The LMS should facilitate the **creation, organization, and delivery of assessments** such as quizzes, exams, and assignments.
   * Teachers should be able to set deadlines for assignments, grade them, and provide feedback.
   * The system must **track student progress**, including the **submission status** of assignments, grades, and the overall performance of students.
   * There should be **automated grading** for objective assessments (e.g., multiple-choice quizzes), while subjective assignments (e.g., essays) should allow for manual grading by teachers.

**Non-Functional Requirements:**

1. **System Requirements:**
   * The system must require **audio and video drives** for multimedia-based content such as lectures, presentations, and other course materials.
   * A **graphics card** is needed to support visual components such as course presentations, interactive media, and video conferencing (if the LMS supports live sessions).
   * The system should ensure proper performance with minimal lag during multimedia usage to enhance the learning experience.
2. **Accuracy of Output:**
   * The LMS must ensure **accurate output** across all processes, including:
     + Correctly displaying course content.
     + Ensuring that assessment results are calculated and displayed properly.
     + Providing accurate grades and feedback for students.
     + Tracking user data accurately (e.g., course progress, assignment submission status).
   * The system must be **reliable**, ensuring that all user interactions lead to the expected results (e.g., submitting assignments, viewing grades, etc.).
3. **Performance and Scalability:**
   * The LMS should handle a large number of users simultaneously without any significant performance degradation.
   * It should **scale effectively** as the number of users, courses, or data grows. This includes handling multiple course registrations, large assignments, and assessments without slowdowns.
4. **Usability:**
   * The system should have an **intuitive interface** that allows users to easily navigate between courses, assignments, and user settings.
   * It should support **responsive design**, ensuring a consistent user experience across various devices such as desktops, tablets, and smartphones.
5. **Security:**
   * The LMS should **protect user data** through encryption methods (e.g., SSL) for login credentials, grades, assignments, and other sensitive information.
   * User roles and permissions should be managed effectively to ensure that users only have access to the features relevant to their role (e.g., students should not have access to course management features).
6. **Backup and Recovery:**
   * The LMS must have a **robust backup system** in place to prevent data loss in case of hardware failure or other issues.
   * The system should have **automatic backup schedules** for user data, course content, and other essential information.