

Online Learning Platform for Skill Development

1. User Authentication:

- Implement a secure login system for different user roles, such as Learner, Instructor, Admin, and Support.
- Use Role-Based Access Control (RBAC) to manage permissions, allowing learners to access courses, instructors to upload content, and admins to manage platform settings.
- Create a Users table with fields: UserId, Username, PasswordHash, Role, Email, and IsActive.

2. Course Management:

- Allow instructors to create and manage courses, including course content, quizzes, and assignments.
- Provide tools for categorizing courses by skill, difficulty level, and type of learning (e.g., video, reading material, or interactive).
- Use a Courses table with fields: CourseId, Title, Description, InstructorId, Category, DifficultyLevel, Duration, and Status.

3. Learning Paths and Certifications:

- Allow learners to follow structured learning paths that guide them through a series of courses to develop specific skills.
- Offer certifications upon completion of learning paths or individual courses to motivate learners and validate their progress.
- Use a LearningPaths table with fields: PathId, Name, Description, CourseIds, CompletionCriteria, and CertificationAwarded.

4. Enrollment and Progress Tracking:

- Enable learners to enroll in courses and track their progress, including completed lessons, quizzes, and assignments.
- Provide tools to allow instructors to track learner progress and give feedback on assignments or exams.
- Use an Enrollments table with fields: EnrollmentId, UserId, CourseId, EnrollmentDate, CompletionStatus, and ProgressPercentage.

5. Quizzes and Assignments:

- Allow instructors to create quizzes and assignments to test learners' knowledge and skills.
- Track quiz results, grades, and provide automatic or manual feedback to learners.
- Use a Quizzes table with fields: QuizId, CourseId, Title, TotalQuestions, Duration, and PassingScore.

- Use an Assignments table with fields: AssignmentId, CourseId, Title, Description, DueDate, and GradingCriteria.

6. Discussion Forums and Peer Interaction:

- Implement discussion forums where learners can ask questions, share insights, and interact with peers and instructors.
- Provide a space for instructors to hold live sessions or office hours for Q&A.
- Use a Discussions table with fields: DiscussionId, CourseId, UserId, PostContent, PostDate, and Status.

7. Content Delivery and Multimedia Support:

- Enable instructors to upload and organize content in various formats, including videos, PDFs, and interactive activities.
- Provide learners with the ability to watch videos, read materials, and complete tasks within the course interface.
- Use a CourseContent table with fields: ContentId, CourseId, ContentType, FileUrl, ContentDescription, and ContentOrder.

8. Feedback and Ratings:

- Allow learners to rate courses and provide feedback on course quality, instructor performance, and platform usability.
- Use feedback to improve future course offerings and help other learners make informed decisions.
- Use a CourseRatings table with fields: RatingId, CourseId, UserId, RatingValue, ReviewComments, and RatingDate.

9. Notifications and Alerts:

- Implement a notification system to alert learners about new courses, upcoming deadlines, quiz results, and announcements.
- Allow users to customize their notification preferences based on course updates, assignments, or system messages.
- Store notifications in a Notifications table with fields: NotificationId, Type, Message, Timestamp, and UserId.

10. Analytics and Reporting:

- Provide analytics tools for both learners and instructors, showing progress, completion rates, and performance in quizzes and assignments.
- Allow admins to generate reports on course enrollment, revenue (if applicable), and learner performance trends.
- Use a Reports table with fields: ReportId, Type, GeneratedDate, Data, and CreatedBy.

11. Instructor Dashboard and Management:

- Provide instructors with a dashboard to manage their courses, view learner progress, grade assignments, and track earnings (if paid).
- Include tools for instructors to communicate with learners, moderate forums, and schedule live sessions.
- Use an InstructorDashboard table with fields: DashboardId, UserId, CourseIds, TotalEarnings, and Messages.

12. Multi-Platform Accessibility:

- Ensure the platform is accessible via desktops, tablets, and mobile devices, allowing learners to access courses on the go.
- Implement offline access for downloadable content, enabling learners to study without an internet connection.
- Design a responsive UI that adapts to different screen sizes, providing a seamless experience for users across all platforms.

Teck Stack

Layer	Technology
BackEnd	.NET 6 or Later with C#
FrontEnd	Angular.js 17 or Later
Database	SQL Server 2022 or MongoDB 6.0
Authentication	JWT or OAuth 2.0

Evaluation Criteria

Criteria	Weightage	Description
Presentation	10 Marks	Clarity, structure, and quality of the presentation, including how well the project is explained.
Code Walkthrough	5 Marks	Clarity of code explanation, organization, and adherence to coding standards.
API Endpoints	25 Marks	Correctness, completeness, and functionality of REST API endpoints with proper HTTP methods.
Frontend	25 Marks	Quality, responsiveness, user experience (UX), and integration with backend APIs.
Models and DB Schema	25 Marks	Accuracy of database schema design, model relationships, and ORM usage.
Innovation	5 Marks	Creative or unique features added beyond the basic requirements.

All the best!!