# Course Management System Technical Documentation

## **1. Technologies Used**

* **Frontend**:
  + **HTML**: Structure and content presentation of web pages.
  + **CSS**: Styling of the user interface.
  + **JavaScript**: Adds interactivity and dynamic features to the pages.
  + **Bootstrap**: Used for responsive design and UI components.
* **Backend**:
  + **ASP.NET Core MVC**: Web application framework for handling requests, responses, and rendering views.
  + **ASP.NET Identity**: Used for managing user authentication, roles, and permissions.
* **Database**:
  + **SQL Server**: A relational database used to store data about users, courses, articles, categories, and other entities.
* **Email Service**:
  + **SMTP**: Simple Mail Transfer Protocol used to send automated email notifications (e.g., for course approvals, rejections).
* **RBA (Role-Based Access)**:
  + Implemented to manage user roles and permissions within the system.

## **2. System Architecture**

### Three-Tier Architecture:

1. **Presentation Layer**:
   * The user interface layer that communicates with users through web pages.
2. **Business Logic Layer**:
   * Handles core business rules such as validating course submissions, user registrations, and sending email notifications.
3. **Data Access Layer**:
   * Manages the interaction with the database to fetch, update, and store data.

## **3. Database Design**

### Tables:

* **Users**:  
  Stores user data, including details such as name, email, role (e.g., "User," "Teacher," "Admin"), and registration status.
* **Courses**:  
  Contains data about courses, including course name, description, duration, approval status, teacher ID, and category ID.
* **Articles**:  
  Stores articles’ data, including title, content, associated images, and the category they belong to.
* **Categories**:  
  Stores course and article categories (e.g., Scientific, Social) for easy filtering.
* **Requests**:  
  Stores requests for teacher registration and course submissions, including their status (approved/rejected/pending).

## **4. User Authentication and Role Management**

* **ASP.NET Identity** is used for handling user authentication and authorization.
  + **Roles**:
    - **User**: Can browse courses, articles, and categories.
    - **Teacher**: Can submit courses for approval and manage approved courses.
    - **Admin**: Can approve/reject courses and teacher registrations and manage content.
* **Role-Based Access**:  
  Different roles are granted different levels of access to system features:
  + **Admin** can access all features and manage all data.
  + **Teacher** has access to their courses and can submit new ones.
  + **User** has read-only access to courses and articles.

## **5. Workflow**

### User Workflow:

1. A user registers and logs in to the system.
2. They browse through available courses and articles.
3. Users can submit course requests for admin review.
4. Once logged in, users can track the approval status of their course submission.

### Teacher Workflow:

1. Teachers register and log in.
2. They can submit new courses for admin approval, including details like name, description, duration, and category.
3. Teachers track their submitted courses’ approval/rejection status.
4. Once approved, they can manage and update their courses.

### Admin Workflow:

1. Admins log in with admin credentials.
2. They can review teacher registrations and course submissions.
3. Admins approve or reject courses based on criteria.
4. Admins manage articles and categories.
5. Admins send email notifications for approval or rejection.