# Cursus Architectural Design Document

##### 1.Introduction

This Architectural Design Document outlines the technical design and implementation details for a web application named Cursus LMS that allows student to register for online courses, and instructor to upload new courses and can get benefits form courses.

##### 2.System Overview

The web application will have the following key features:

User registration and login.

Admin approve to create instructor account

Course catalog with course details and descriptions.

Course registration and enrollment for student.

User dashboard to view registered courses and progress.

Administrative tools to manage courses, instructors, and student enrollments.

Buy courses using payment.

##### 3. Technology Stack

The web application will be built using the following technologies:

Front-end: React.

Back-end: ASP.NET Core

Database: Microsoft SQL Server

Storage: Storage Firebase

Hosting: Docker, Azure

##### 4.High-Level Architecture

The web application will follow a standard 3-tier architecture, consisting of:

Presentation Layer: The front-end user interface built with swagger to test API function.

Application Layer: The back-end server built with Node.js and Express.js, responsible for handling business logic, API endpoints, and database operations.

Data Layer: The SQL Server database, storing user accounts, course details, enrollment information, and other relevant data.

The overall architecture can be illustrated as follows:

##### 5.Key Components

The web application will consist of the following key components:

User Management for admin

User registration and login

User profile management

Password reset functionality

Course Catalog.

Course listing with details and descriptions

Course search and filtering

Course enrollment.

Enrollment Management.

Course registration and enrollment

User dashboard to view registered courses.

Progress tracking and certification for student.

Administrative Tools.

Course management (create, update, delete).

Instructor management.

Student enrollment management.

##### 6.Security Considerations:

The web application will implement the following security measures:

Role-based access control (RBAC) for administrative functions

Secure communication between the front-end and back-end using HTTPS

Input validation and sanitation to prevent SQL injection and XSS attacks

Regular security audits and updates to address any vulnerabilities

##### 7.Scalability and Performance

To ensure the web application can handle increasing user load and data volume, the following strategies will be employed:

Horizontal scaling of the application servers using a load balance.

Caching mechanisms for frequently accessed data (e.g., course catalog)

Optimized database queries and indexing

##### 8. Deployment and Hosting

The web application will be hosted on Azure and Docker, a popular cloud platform for deploying and scaling modern web applications. The application will be continuously deployed from the main Git-lab repository, ensuring seamless updates and rollbacks.

##### 9. Database Design















