# Module 10 Assignment: Project Architecture and DevOps Basics – Transcript GitHub Repository: <https://github.com/jagadeeshm007/lms_module10>

## Objective

The objective of this assignment is to equip students with skills to design a scalable project architecture using the MERN stack. Students will also implement basic DevOps practices, including setting up a Git repository and understanding CI/CD pipelines. Optional tasks include exploring containerization with Docker to enhance deployment workflows.

## Scenario

You are now finalizing the foundational architecture for the Learning Management System (LMS). This includes structuring the project for scalability and maintainability, setting up version control with Git, and preparing for deployment through CI/CD pipelines. A well-structured project and DevOps integration will ensure the LMS can handle future enhancements and deployments efficiently.

## Requirements

1. 1. Project Architecture

* Structure a Scalable MERN Stack Project:

- Separate the backend (Node.js/Express) and frontend (React) into distinct directories (/backend and /frontend).  
- Create reusable services for database operations in the backend.  
- Use context or state management for shared data in the frontend.  
- Store sensitive data (e.g., database URL, API keys) in .env files for both backend and frontend.  
- Ensure environment variables are correctly loaded and used.  
- Implement Separation of Concerns (SoC).  
- Add Configurable Environment Variables.

1. 2. DevOps Basics

* Understand CI/CD Pipelines:

- Write a basic YAML configuration file for a CI/CD pipeline using GitHub Actions to:  
 - Run tests for the backend and frontend.  
 - Build the React application.  
 - Include at least one automated test for the backend (e.g., testing an API endpoint) and one for the frontend (e.g., testing a component).  
- Create a Dockerfile for the backend to containerize the Node.js application.  
- Use docker-compose.yml to run both backend and frontend containers.  
- Set Up Automated Tests.  
- Optional: Basics of Containerization with Docker.

1. 3. Version Control with Git

* Set Up a Git Repository:

- Initialize a Git repository for the LMS project.  
- Create a .gitignore file for excluding unnecessary files (e.g., node\_modules, .env).  
- Use the following branching strategy:  
 - main for stable releases.  
 - development for ongoing features.  
 - Feature-specific branches (e.g., feature/file-upload).  
- Write clear and meaningful commit messages.  
- Ensure each commit represents a logical change.  
- Create a public or private repository on GitHub.  
- Push the local project to GitHub and configure branch protection rules for main.  
- Organize Workflows with Branching.  
- Commit Changes.  
- Push the Repository to GitHub.