Project Proposal

Course: Transactional Web Applications
Project Title: Comparative Analysis of Deployment Strategies and Architectures for an E-Commerce Platform
Student: Reza Akbari
Date: 04/07/2025

🔹 1. Project Objective

The aim of this project is to analyze and compare two different technology stacks and deployment strategies for a transactional web application. Using the developed **IdealShop** e-commerce platform as a business scenario, the project will evaluate the performance, scalability, usability, and deployment efficiency of two distinct web stacks: **ASP.NET Core with SQL Server** and **Node.js with MongoDB**.

🔹 2. Business Scenario

IdealShop is an online shopping platform where customers can:

- Register and log in
- Browse products by category
- Add items to a shopping cart
- Place orders securely with a checkout form

Admins can:

- Log in to a control panel
- Create, update, and delete products and categories
- View and manage customer data

This real-world use case provides a suitable base for comparing backend and frontend technologies, database performance, and hosting strategies.

🔹 3. Hypotheses (PoC Statements)

1. **Performance Hypothesis**: ASP.NET Core with SQL Server performs better under concurrent transactional load than Node.js with MongoDB.

2. **Deployment Hypothesis**: Dockerized deployment is easier to scale and maintain compared to traditional hosting models.
3. **Cost Hypothesis**: Azure App Service provides a better cost-performance ratio for transactional workloads than AWS Elastic Beanstalk.

🔹 4. Proposed Stacks for Comparison

Component	Stack A (Microsoft Stack)	Stack B (Open Source Stack)
Backend	ASP.NET Core MVC	Node.js with Express.js
Frontend	Razor Pages	React.js
Database	SQL Server	MongoDB
Hosting	Azure App Service	AWS EC2 or Heroku (Free tier)
Deployment	Docker + Azure Container Apps	Docker + AWS Lightsail/Heroku

---

🔹 5. Deliverables

- Two prototypes of IdealShop using different tech stacks
- Dockerized versions of both prototypes
- Performance and usability testing results
- Comparative analysis in final report
- GitHub repo with full source code
- PDF report with summary of findings and recommendation

🔹 6. Timeline

Week	Task
1	Write proposal and define architecture
2	Build Stack A prototype (ASP.NET)
3	Build Stack B prototype (Node.js)

Week	Task
4	Containerize both apps using Docker
5	Deploy to Azure and AWS
6	Run performance and usability tests
7	Analyze results and finalize report
8	Submit report and presentation