TechXpress E-commerce Platform - Project Plan

Project Overview

Objective:

Develop a scalable and maintainable e-commerce platform for selling electronics using ASP.NET Core.

It will implement design patterns like NTier Architecture, Repository Pattern, and Unit of Work.

Key Features:

- User-friendly browsing and shopping experience.

- Integrated payment system (Stripe).

- Admin panel for product, category, and order management.

- Role-based access control.

- Deployment on Microsoft Azure or Hostinger.

Technologies & Tools

- Backend: ASP.NET Core MVC, Entity Framework Core, ASP.NET Identity

- Frontend: JQuery, DataTables, Bootstrap, Toaster JS

- Payment Integration: Stripe

- Deployment: Microsoft Azure/Hostinger

Design Patterns

1. NTier Architecture: Presentation, Business Logic, and Data Access layers.

2. Repository Pattern: Abstracts data access.

3. Unit of Work Pattern: Ensures atomic transactions.

4. Dependency Injection: Promotes loose coupling and testability.

TechXpress E-commerce Platform - Project Plan

4-Week Development Plan

Week 1: Initial Setup & Product Listings

Tasks:

- Set up the NTier architecture:

  - Presentation Layer: ASP.NET MVC with views, controllers, JQuery, DataTables.

  - Business Logic Layer: Manage business rules (e.g., product addition, order calculations).

  - Data Access Layer: Implement repositories and Unit of Work for data management.

- Database Design: Use Entity Framework Core to define entities like Products, Categories, Orders, and Users.

- Authentication Setup: Integrate ASP.NET Identity for registration, login, and roles.

Deliverables:

- Complete NTier setup with functional layers.

- Basic product listing page with search and filters.

- Implemented database schema with repository and Unit of Work patterns.

- User authentication system.

Week 2: Shopping Cart, Role-Based Access, and Admin Panel

Tasks:

- Shopping Cart: Use Session State to store and manage cart data.

- Role Management: Define Customer and Admin roles with restricted access.

- Admin Dashboard: Implement product, category, and order management using DataTables.

Deliverables:

- Working shopping cart with product addition and review features.

- Role-based access control system.

- Admin dashboard with CRUD operations for products and orders.

Week 3: Order Placement, Payments, and User Profiles

Tasks:

- Order Placement: Ensure atomic transactions for order processing.

- Stripe Integration: Implement secure payment processing.

- User Profiles: Enable users to view order history and manage profiles.

Deliverables:

- Integrated order placement with Stripe payments.

- Functional customer profile and order history section.

- Thoroughly tested transaction process.

Week 4: UI Enhancements, Testing, and Deployment

Tasks:

- UI Enhancements: Polish the UI using Bootstrap and custom CSS.

- DataTables Enhancements: Enable sorting, filtering, and pagination for admin management.

- Final Testing: Conduct end-to-end testing for the entire system.

- Deployment: Deploy to Microsoft Azure/Hostinger and finalize documentation.

Deliverables:

- Responsive and polished UI for frontend and admin panels.

- DataTables functionalities fully implemented.

- Fully tested and deployed application.

- Completed documentation for setup and usage.

Risk Management

- Delays in Payment Integration: Start early and ensure multiple testing phases.

- Deployment Issues: Prepare contingency plans with detailed documentation and testing.

- Role-Based Access Bugs: Conduct rigorous testing for various role scenarios.

Next Steps

- Set up the initial development environment.

- Define the database schema and relationships.

- Break down tasks into detailed sprints for better tracking.