E-commerce Project Plan

# 1. Project Overview

Project Name: E-commerce Platform

Team Structure:

* Backend Team: Focus on API development, database integration, and business logic.
* Frontend Team: Develop the user interface using a framework like React.
* Mobile Team: Create a mobile application using Flutter or React Native.
* Testing Team: Ensure the application is bug-free and meets quality standards.

# 2. Define Project Requirements (Day 1)

Functional Requirements:

* User Authentication and Authorization.
* Product Management (CRUD operations).
* Shopping Cart and Checkout process.
* Payment Integration.
* Order Management.
* User Profile Management.

Non-Functional Requirements:

* Scalability.
* Security (SSL, data encryption, etc.).
* Performance.
* User-friendly UI/UX.
* Mobile Responsiveness.

# 3. Team Assignments (Day 1)

Backend Team:

* Design the database schema using PostgreSQL.
* Develop REST APIs using Spring Boot.
* Implement authentication (JWT).
* Set up role-based access control.

Frontend Team:

* Design the UI/UX.
* Implement the frontend using React.
* Integrate with backend APIs.

Mobile Team:

* Develop the mobile app UI.
* Integrate with backend APIs.
* Implement push notifications and payment integration.

Testing Team:

* Write test cases for each module.
* Perform unit testing, integration testing, and end-to-end testing.
* Ensure cross-browser compatibility for the frontend.

# 4. Set Milestones (Day 1)

Day 1-2: Requirement Analysis, Database Design, and API Planning.

Day 3-4: Backend and Frontend Development.

Day 5: Mobile Application Development.

Day 6: Integration and Testing.

Day 7: Final Review, Bug Fixes, and Documentation.

# 5. Daily Check-ins (Day 1-7)

Morning Stand-up: Discuss what was done yesterday, today’s tasks, and any blockers.

End-of-Day Review: Ensure milestones are on track and adjust the plan as needed.

# 6. Documentation (Continuous)

Technical Documentation:

* Backend: API documentation, database schema, security implementation.
* Frontend: Component structure, state management, and third-party libraries used.
* Mobile: App architecture, API integration, platform-specific code.
* Testing: Test cases, test plans, and reports.

User Documentation:

* User guides for the application (web and mobile).
* Admin panel documentation.

# 7. Final Presentation (Day 7)

Prepare a Presentation:

* Overview of the project.
* Key features and functionalities.
* Demo of the application.
* Challenges faced and how they were resolved.
* Future improvements.