Faculty of Computers and Artificial Intelligence

Beni-Suef University

E-Commerce

Presented By:

1. Abdel-Rahman Waleed Ahmed
2. Omar Yasser Abdelkhalk
3. Mohamed Eid Sayed
4. Abdel-Rahman Maged Qarni

Supervised By

Eng: Abdel-Rahman Hesham Noyer

DEPI (Front-end) || (React) : 2024-2025

Table of Contents

Chapter One
(Introduction)

Abstract

The E-commerce Platform is a web-based system designed to simplify the process of buying and selling products online.
It allows users to register, browse products, add items to a shopping cart, make secure payments, and track their orders easily.
The project focuses on creating a user-friendly interface, efficient product management, and a secure transaction process to enhance the overall online shopping experience.

Overview

Today's digital world, e-commerce platforms have become essential for connecting customers and businesses globally.
This project provides an interactive and responsive website where users can explore products, view detailed information, and complete purchases without physical interaction.
The system integrates several modules including user authentication, product catalog, cart management, order processing, and payment gateway integration.

Problem Definition

Traditional shopping methods often require physical presence, which consumes time and effort.
Many small businesses also face challenges in reaching a wider audience due to limited resources.
This project aims to solve these issues by offering an online platform that allows customers to shop conveniently anytime, anywhere — while helping sellers manage and promote their products efficiently

Project Goal

The main goal of the E-commerce Platform is to build a secure, reliable, and easy-to-use online shopping system.
It aims to enable customers to:

- Browse and search for products easily.
- Add items to a cart and purchase them securely.
- Track orders and manage their account effortlessly.
  Additionally, the system provides an admin interface to manage products, users, and transactions efficiently.

App Features

- **User Registration & Login** – Secure user authentication and account management.
- **Product Browsing** – Search, filter, and view detailed product information.
- **Shopping Cart** – Add, edit, or remove items with real-time price updates.
- **Checkout & Payment** – Multiple payment methods with order confirmation.
- **User Account Management** – View and update personal data and order history.
- **Order Tracking** – Monitor the status of current and past orders.
- **Logout** – End session securely to protect user data.

Purpose

The purpose of this project is to develop a comprehensive and efficient e-commerce system that bridges the gap between customers and retailers.
It provides a digital solution for users who prefer convenience, accessibility, and security in their shopping experience.
Through this platform, both users and administrators benefit — users enjoy an easy shopping process, and administrators gain a centralized way to manage products and transactions effectively.

Chapter Two
(Methodology)

Tools&Environment

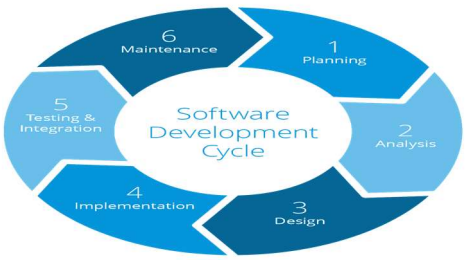
The development of the E-commerce Platform was carried out using modern and efficient technologies to ensure performance, scalability, and maintainability.
The tools and environment used are as follows:

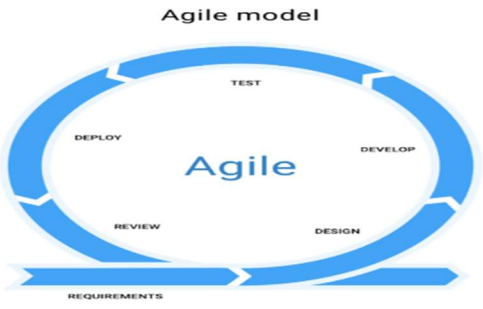
- **Frontend:** HTML5, CSS3, JavaScript, and Bootstrap for a responsive and user-friendly interface. And React
- **Backend:** Node.js for handling server-side operations and data processing.
- **Database:** MySQL for structured data storage and fast retrieval.
- **Frameworks:** React.js (optional) to enhance the dynamic behavior of the user interface.
- **IDE:** Visual Studio Code for coding and debugging.
- **Version Control:** Git and GitHub for managing versions and collaboration.
- **Testing & Deployment:** XAMPP for local testing

Project Methodology

The project follows the **Agile Scrum Methodology**, which focuses on iterative and incremental development.
In this approach, the project is divided into short, manageable phases called **Sprints**, where each sprint delivers a functional part of the system — such as login, cart management, or payment processing.

The Scrum framework involves continuous collaboration between developers and stakeholders, ensuring that every feature is reviewed, tested, and improved before moving forward





Scrum Roles

- **Product Owner:** Defines goals and priorities (stakeholders).
- **Scrum Master:** Oversees sprint progress and removes obstacles.
- **Development Team:** Implements, tests, and integrates the features.

Advantages of Scrum

- **Enhances teamwork and communication.**
- **Ensures early testing and continuous feedback.**
- **Increases adaptability to changes in requirements.**
- **Provides transparency and measurable progress.**
- **Delivers working modules quickly for review and validation**

Project planning timeline

Phase	Duration	Main Activities
1	Week 1–2	Requirement gathering, UML diagrams, system design
2	Week 3–4	Frontend development (Login, Homepage)
3	Week 5–6	Backend logic, database integration
4	Week 7–8	Cart, Checkout, and Payment gateway implementation
5	Week 9–10	Testing, debugging, and deployment

Project planning

The project planning phase ensures that every task is organized and tracked efficiently. It involves defining the project scope, objectives, milestones, and responsibilities. The major planning steps include:

- Collecting and analyzing system requirements.

- Designing the architecture and database schema.
- Dividing the project into modules (frontend, backend, database).
- Assigning each module to a sprint.
- Integrating and testing all modules before final deployment.

Effective planning ensures smooth coordination and timely completion of the project.

Requirements

ID	Requirements	Date	Related Features	Related Stakeholders
1	**User Registration & Login**	2025-10-6	Login Page – Authentication – Validation	Customer – System Administrator
2	**Homepage & Product Browsing**	2025-10-6	Homepage – Product List – Search & Filter	Customer – Marketing Team
3	**Product Details**	2025-10-6	Product Details Page – Product Info – Image Viewer	Customer – Content Manager
4	**Shopping Cart**	2025-10-6	Cart Page – Update/Delete Items – Price Calculator	Customer – System Developer
5	**Checkout & Payment**	2025-10-6	Checkout Page – Payment Gateway – Email Notification	Customer – Finance Department
6	**User Account Management**	2025-10-6	Account Page – Profile Update – Order History	Customer – Admin Support
7	**Order Management**	2025-10-6	Orders Module – Status Tracking – Notifications	Customer – Warehouse Staff – Admin

Chapter Three

(System Architecture)

System analysis and design

The system architecture for the **E-commerce Platform** serves as a comprehensive blueprint that transforms a business concept into an interactive, user-friendly, and scalable online shopping experience.
By leveraging **Unified Modeling Language (UML)** techniques, the analysis and design phase provides a structured, visual understanding of how different system components—such as user interfaces, databases, and payment modules—interact with one another before the development begins.

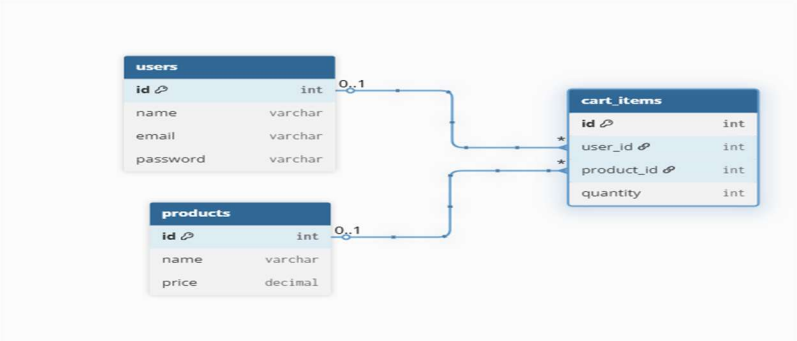
UML diagrams including **Use Case, Flowchart, ERD, DFD, Sequence, and Class diagrams** ensure a clear representation of user behaviors, data processing, and system logic. This systematic approach enables developers and stakeholders to identify requirements, eliminate ambiguities, and design an efficient, secure, and reliable architecture.

Ultimately, the **E-commerce System Design** supports seamless navigation, secure transactions, and real-time data flow between users, products, and orders—resulting in a robust and scalable online marketplace that

Using UML techniques, we built:

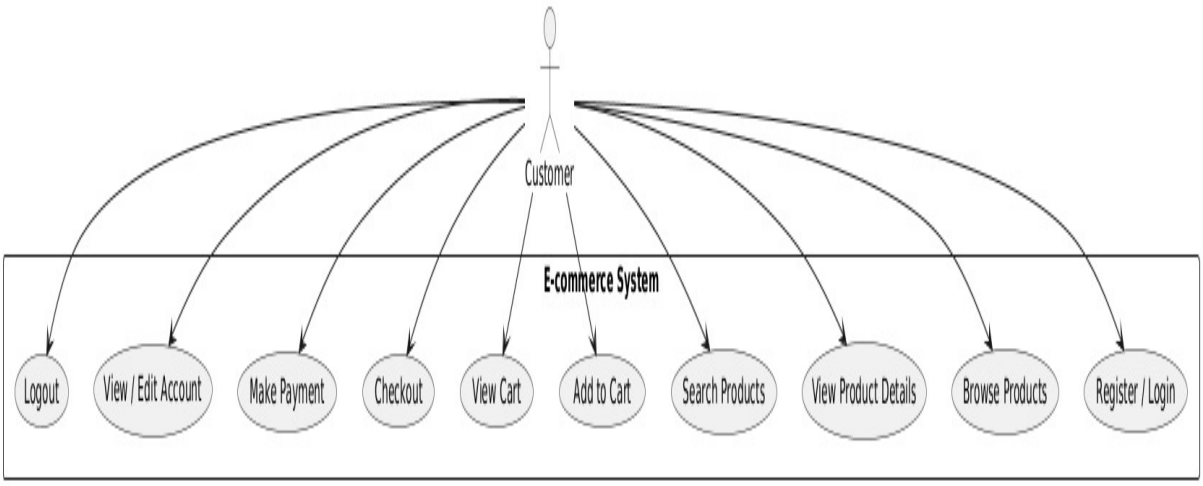
- **Database Schema**
- **Use cases**
- **ERD**
- **User scenario**
- **Flowchart**
- **Class diagram**
- **Sequence**
- **DFD**

Database Schema

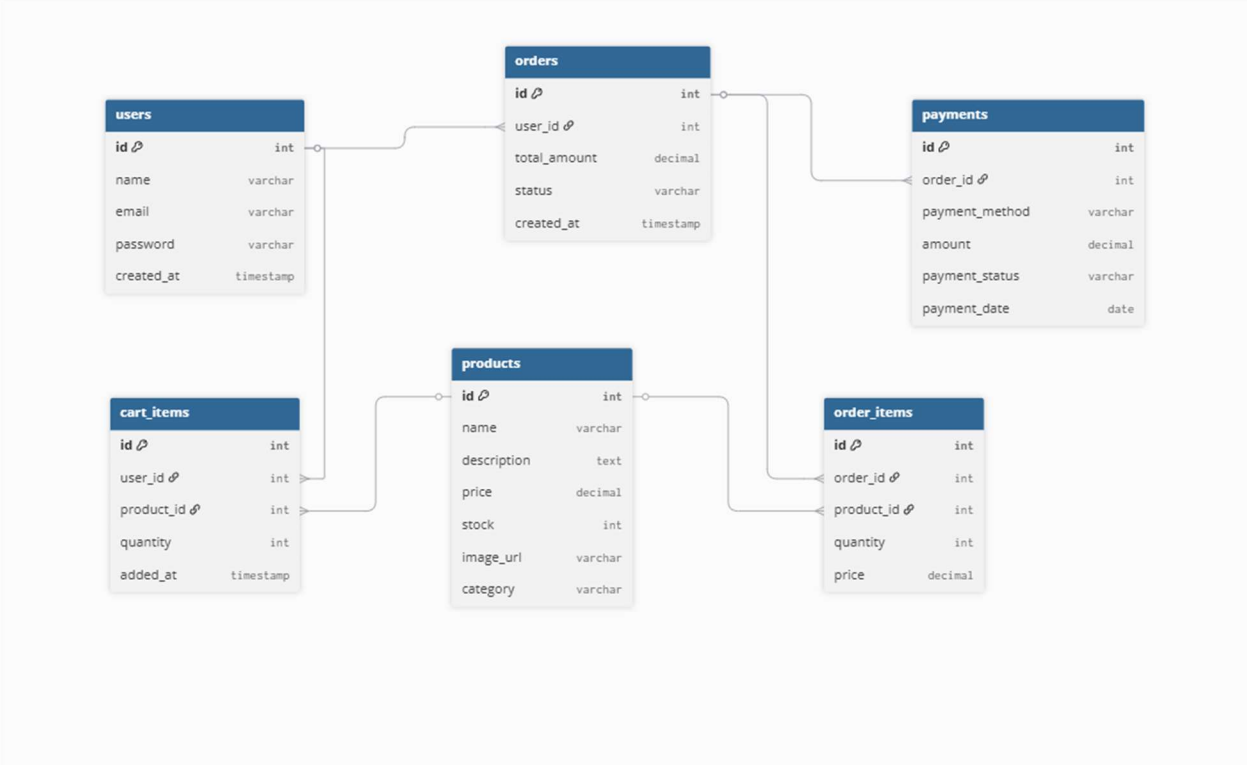


Use Case diagram

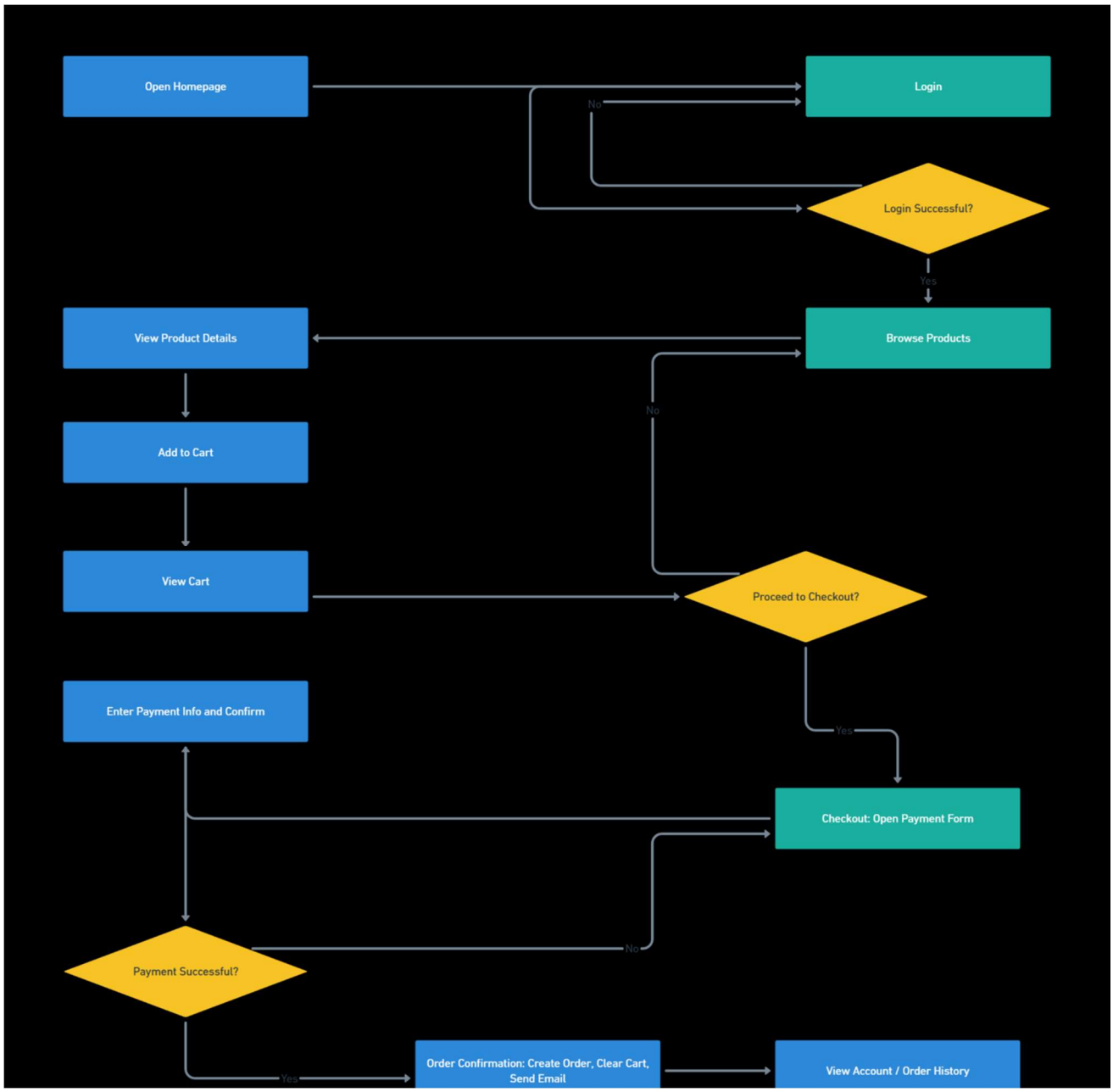
The Use Case Diagram captures the interactions between actors (users, developers, technical support) and the system, highlighting key functionalities.



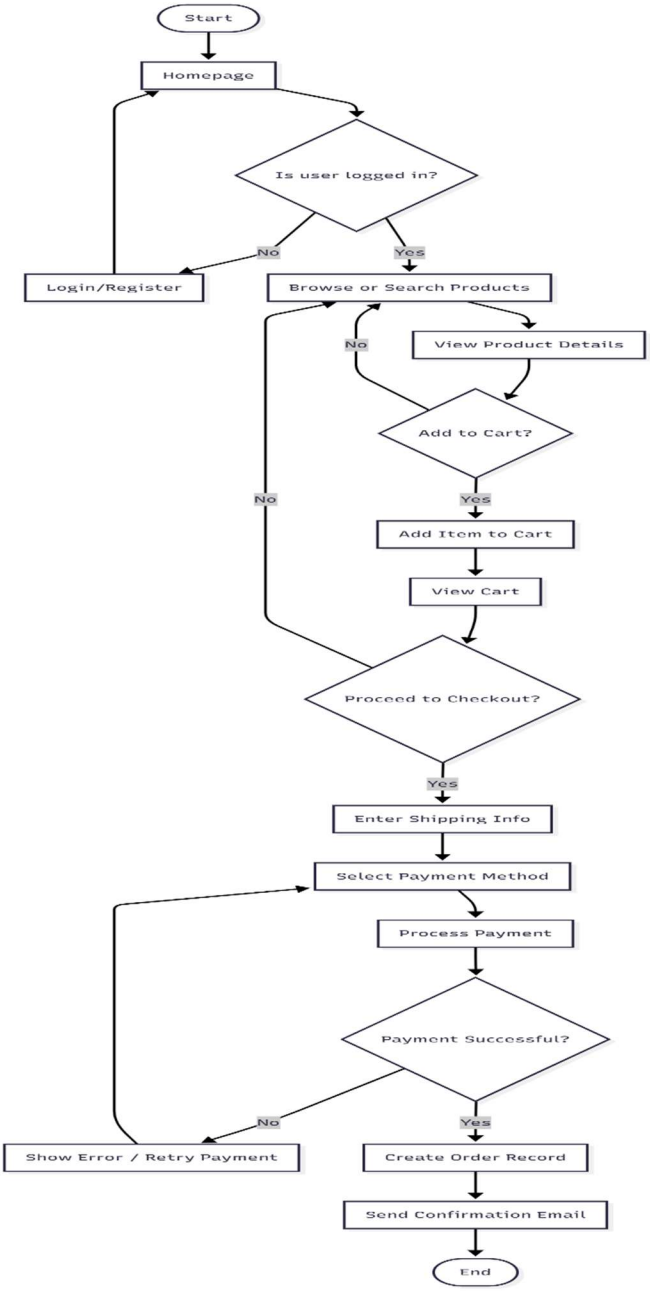
ERD Diagram



User Scenario



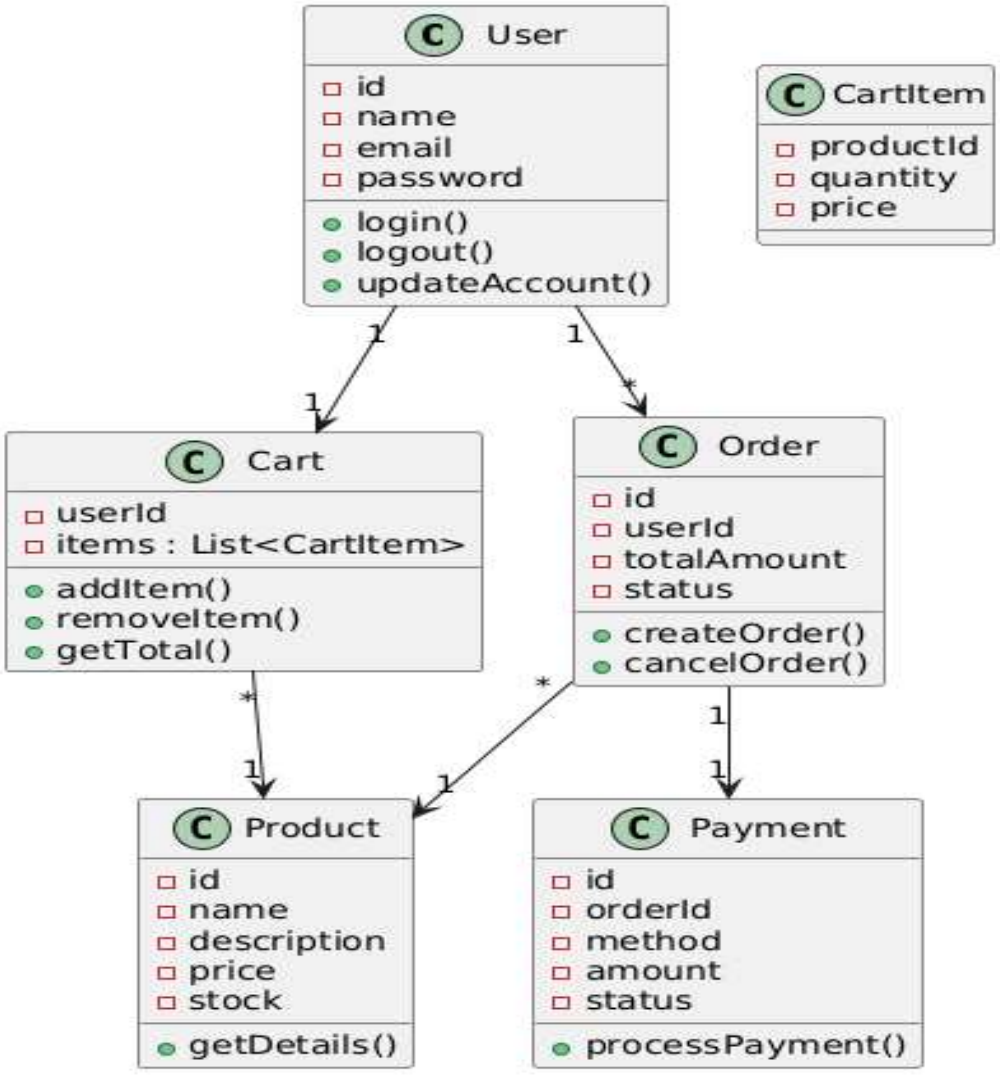
Flow Chart Diagram



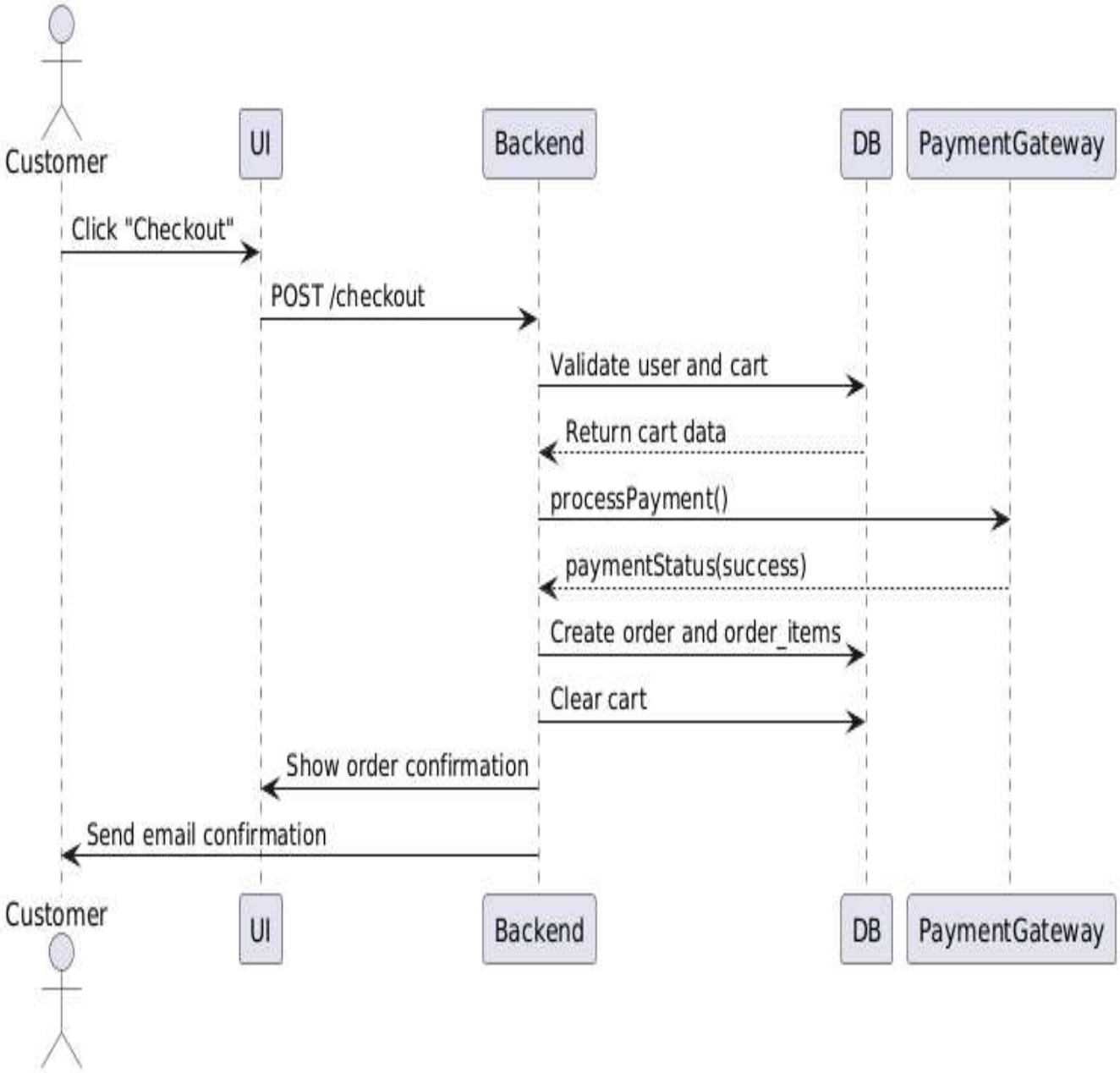
Class Diagram

Class diagram represents the structure and relationships of a system's classes and their interactions. It provides a static view of the system, focusing on the classes, attributes, operations, and associations between classes. Class diagrams depict a static view of the model, or part of the model, describing what attributes and behavior it has rather than detailing the methods for achieving operations. Class diagrams are most useful in illustrating relationships between classes and interfaces.

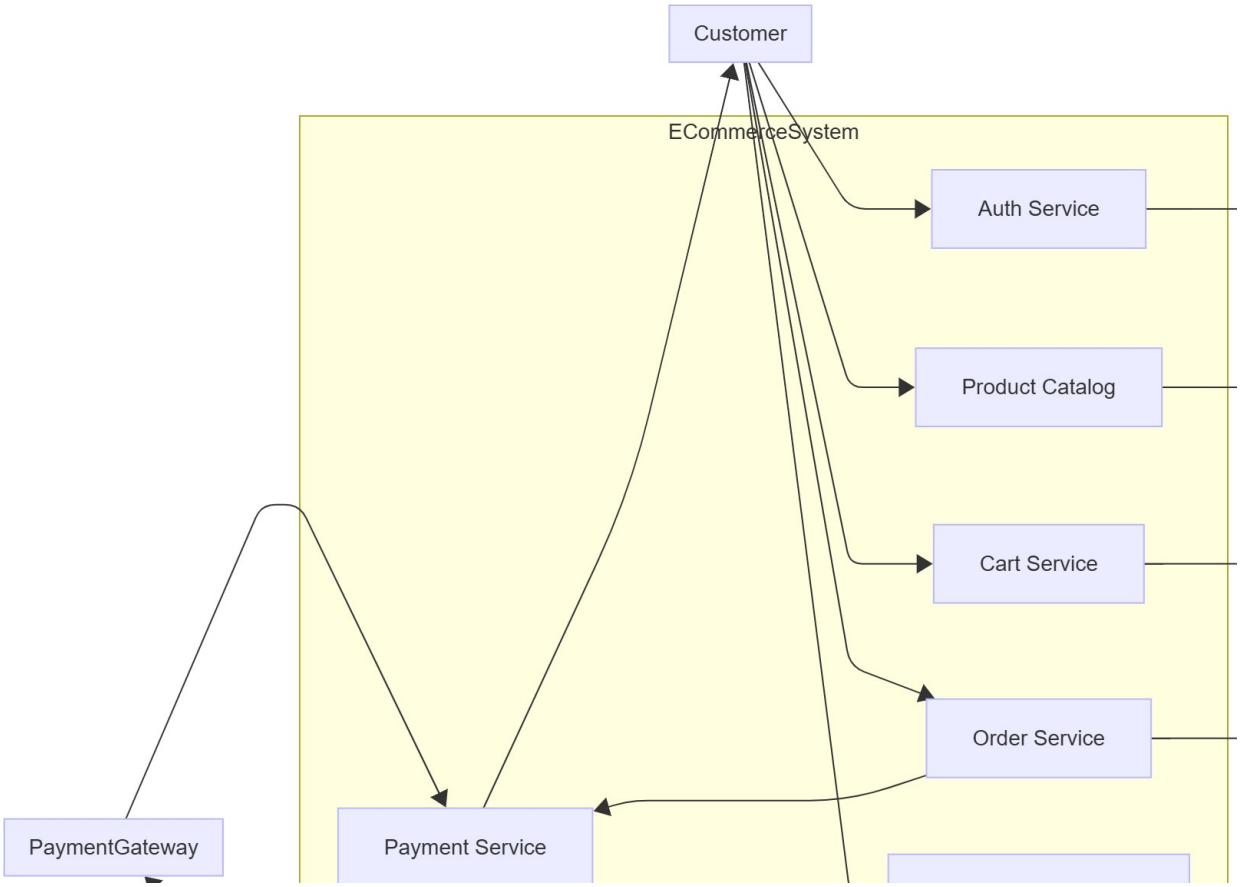
- This class diagram shows the user 's interaction with the system.
- Users can view the home page.



Sequence Diagram

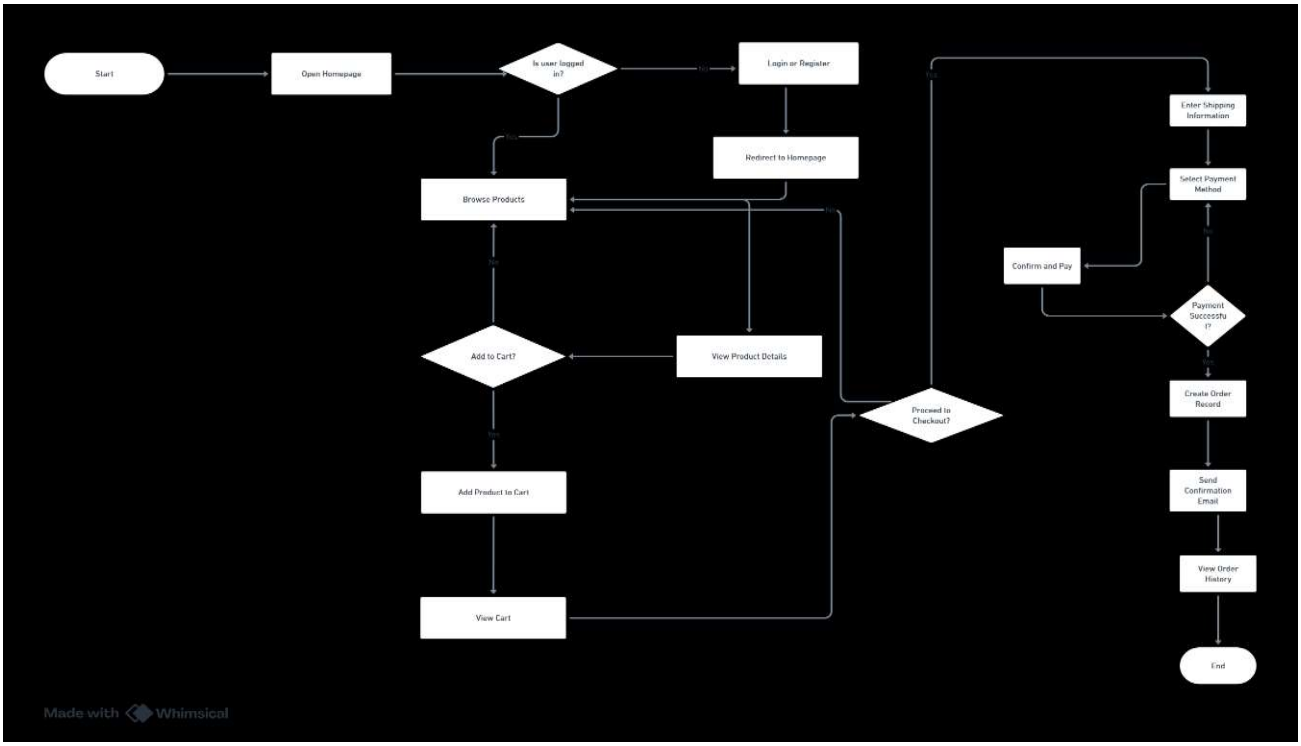


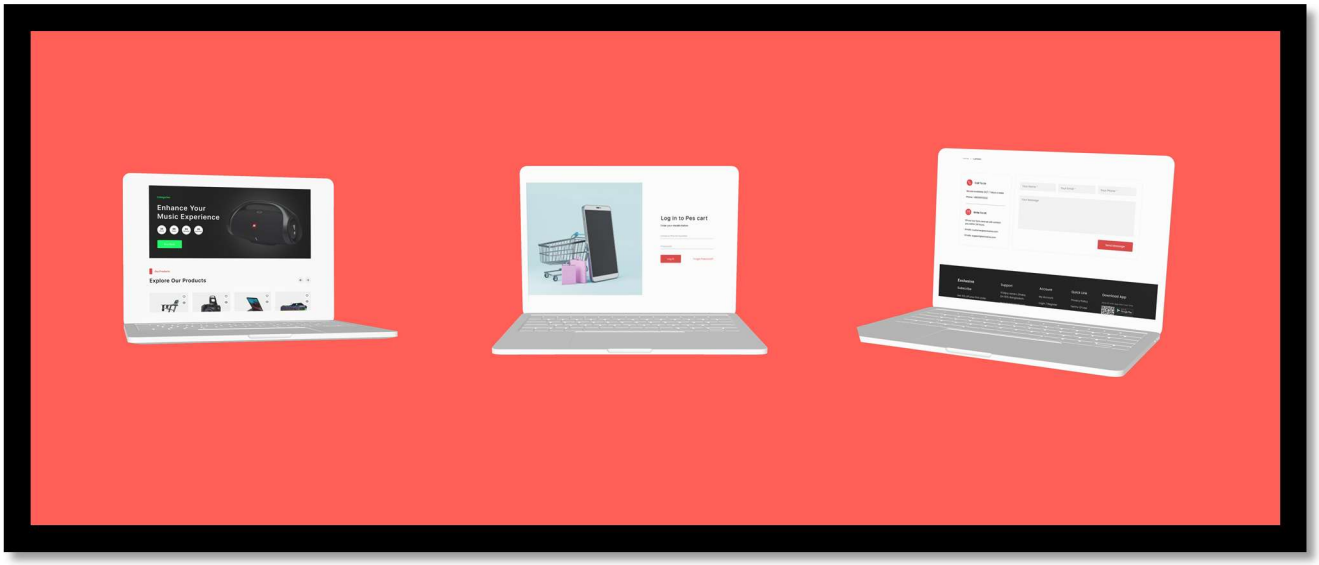
Data Flow Diagram

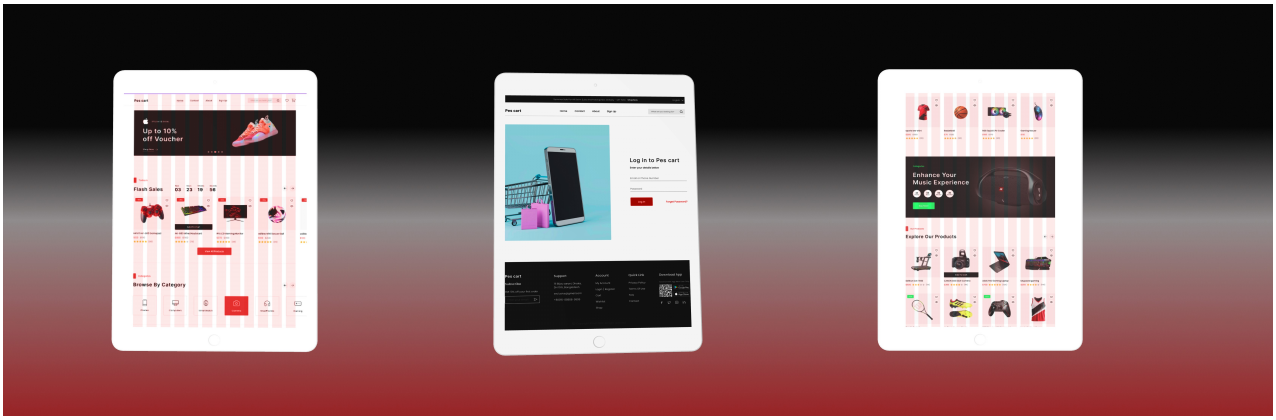


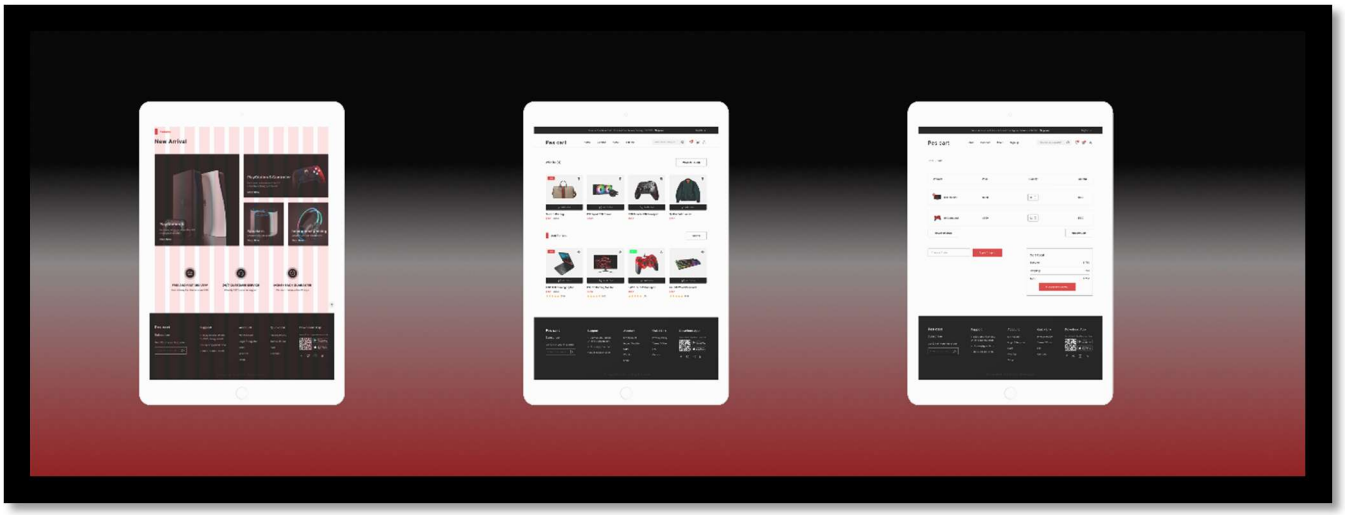
Chapter Four
(UI &UX)

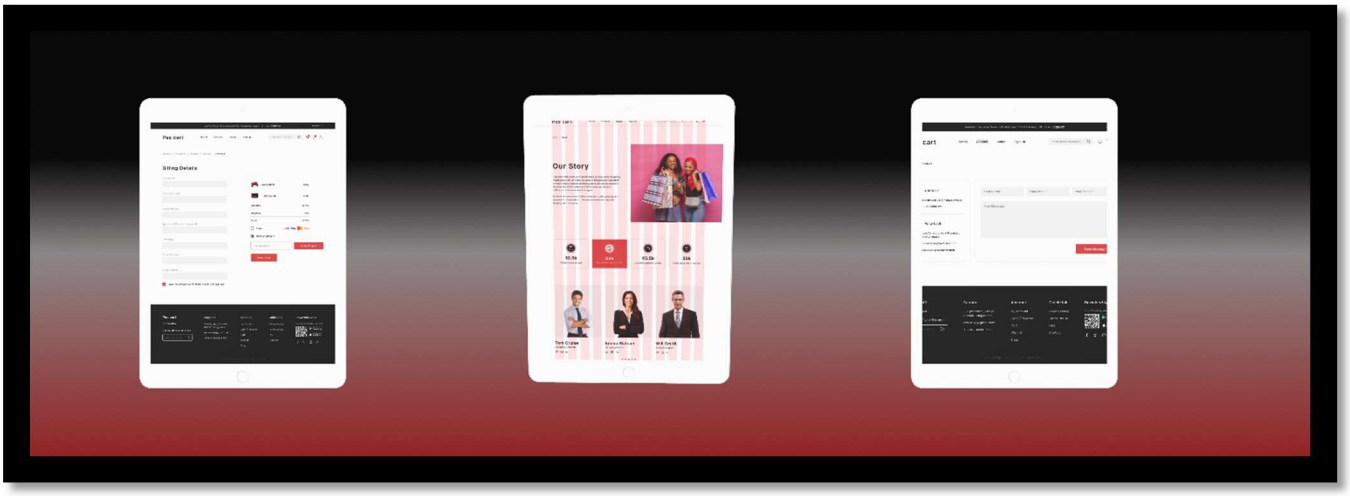
User flow for Project:






Up to 10%
off Voucher
Flash Sales
Browse By Category
Log in to Pes cart
Enhance Your
Music Experience
Explore Our Products


New Arrival
Pes cart


Billing Details
Our Story
cart

Chapter six
(Front -End)

React

Framework

WhyReact?

React is a powerful and widely-used JavaScript library developed by Facebook for building dynamic and responsive user interfaces, especially for single-page applications (SPAs). It provides developers with a component-based architecture that encourages reusable UI elements, simplifying the process of creating complex user interfaces.

With the help of modern tools and libraries, the React ecosystem becomes even more robust:

- **Vite** offers a super-fast development environment with lightning-quick hot module replacement (HMR), optimized build output, and native ESM support.
- **React Router DOM** allows seamless navigation and route management for SPAs.
- **React Hook Form** simplifies the process of building and validating forms with minimal re-renders.
- **Redux Toolkit** streamlines Redux state management, reducing boilerplate and encouraging best practices.
- **Yup** enables powerful and schema-based form validation, often used in combination with React Hook Form.
- **Axios** provides a promise-based HTTP client for API communication and request handling.

React's declarative syntax and virtual DOM make it fast and developer-friendly, leading to a better developer experience and more maintainable codebases.

Project Structure

The project structure is organized with maintainability, scalability, and mod

Chapter seven
(Backend)

اللهم إن أهلنا في غزة قد ضاقت بهم السبل، واشتدّ عليهم البلاء، فكن لهم عونًا ونصيرًا،

اللهم اجبر كسرهم، وارحم ضعفهم، وآمن روعاتهم، وانصرهم على من ظلمهم.

اللهم احفظهم بحفظك، وارفع عنهم الحصار والدمار، وبدّل خوفهم أمنًا، وحزنهم فرحًا،

اللهم عجّل لهم بالفرج، وانصرهم نصرًا مؤزرًا، فأنت القوي الجبار، وأنت حسبنا ونعم الوكيل.

THANKS