COLLEGECODE : 9528

COLLEGENAME : SCAD COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

STUDENTNMID: C90AAB851852669AF3C40AA5C0072C14

Roll no : 952823104163

DATE : 26.09.2025

Completed the project named as:

Phase 3

TECHNOLOGY PROJECT NAME :

E COMMERCE PRODUCT PAGE

SUBMITTED BY,

NAME : M.SUGANYA

MOBILE NO : 8754893885

Phase2- MVP Implementation

1.Project Setup

starting

Bdevfoerloe pmentthe MVP implementation, the environment was configured to ensure smooth progress. This includes creating the project structure, installing dependencies, and initializing version control.

Frontend Setup:

- Framework: React.js for building reusable and responsive UI components.
- Dependencies: react-router-dom (routing), axios (API calls), tailwindcss (styling).
- Project Initialization: Used create-react-app for project scaffolding.

Backend Setup:

- Framework: Node.js with Express.js for handling REST API requests.
- Dependencies: express, cors, mongoose (for MongoDB), dotenv (environment variables).
- Folder Structure: Organized into routes, controllers, and models for maintainability.

Environment Configuration:

- .env file created for sensitive credentials (database URL, port number).
- Local server configured:
  - Backend: http://localhost:5000
  - Frontend: http://localhost:3000

This setup ensures seamless integration between frontend and backend, ready for core feature development.

2. Core Features Implementation

The MVP prioritizes essential e-commerce functionality: browsing products, viewing details, and managing a shopping cart.
Implemented Features:

1. Product Listing Page:
   - Displays a grid of products with images, names, prices, and a "View Details" button.
   - Responsive layout for desktop and mobile.
   - Fetches data from backend endpoint: /api/products.
2. Product Details Page:
   - Shows complete product information: description, price, stock availability.
   - Includes an "Add to Cart" button.
   - Dynamic routing example: /product/:id.
   - Data fetched from backend based on product ID.
3 .Shopping Cart Functionality:
   - Add, update, or remove products.
   - Cart state persisted in localStorage.
4. Basic Navigation:
   - Header with links: Home, Cart, Login.
   - Routing implemented using react-router-dom.
5. User-Friend ly UI:
   - Minimal and consistent interface using Tailwind CSS.
   - Styled buttons and forms for a cohesive experience.

These features provide the core user flow: browse → view → add to cart, forming the backbone of the MVP.

3.DataStorage(LocalState / Database)

Local State Management:

- Used React useState and useContext for temporary UI states like cart items and selected product.
- localStorage ensures persistence across page refreshes.

Database Storage:

- Database: MongoDB for scalability and JSON-based structure.
- Collections:
  - products: stores id, name, description, price, image URL, stock availability.
  - carts: stores user cart data with product references and quantities.
- API endpoints /api/products and /api/cart allow frontend-backend interaction.

This hybrid approach ensures the MVP works offline with local data whi le supporting full database integration for real-world scalability.

4.Testing Core Features

Unit Testing:

- React components tested individually (e.g., ProductCard displays correct name and price).

API Testing:

- Postman used to verify endpoints (/api/products, /api/product/:id, /api/cart).
- Checked correct data response and CRUD operations.

Manual Testing:

- Verified complete user flows: browse → view → add to cart → checkout preparation.
- Identified and fixed UI/functional bugs.

Deliverable: Stable, fully functional MVP ready for demonstration.

5. Version Control (GitHub)

- Initialized Git repository and pushed initial project.
- Branching strategy:
  - main for stable builds
  - dev for active development
- Frequent commits with meaningful messages.
- Pull requests and code reviews maintained for team collaboration.

Deliverable: Project fully tracked on GitHub with version history and collaboration workflow in place.