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 2

TECHNOLOGY PROJECT NAME :

E COMMERCE PRODUCT PAGE

SUBMITTED BY,

NAME : M.SUGANYA

MOBILE NO : 8754893885

Phase2- Solution Design &architecture

1.Tech Stack Selection

Toensurescalability, performance, and ease of maintenance, the following technologies are chosen:

- Frontend → React.js
- Provides a fast, responsive, and component-based UI for the product page.
- Backend → Node.js with Express
- Used to build REST APIs for products, cart, wishlist, and reviews.
- Database → MongoDB
- A NoSQL database with a flexible schema, suitable for handling products, user carts, and reviews.
- Authentication → JWT (JSON Web Tokens)
- Ensures secure login and session management for users.
- Deployment → Vercel for frontend hosting and AWS/Heroku for backend services.

2. UI Structure / API Schema Design

UI Structure (Main Components):

- ProductPage → Parent container for the page.
- ProductImageGallery → Displays product images in a slider.
- ProductInfo → Shows product title, description, price, and stock status.
- AddToCartButton → Lets users add products to cart.
- WishlistButton → Saves product for later.
- RatingsAndReviews → Shows average rating and customer reviews.
- RelatedProducts → Suggests similar products.

Sample Product API Schema:

{"productId": "P12345","name": "Wireless Bluetooth Headphones","price": 1999,"description": "High-quality wireless wheitahd phones noise cancellation.","images": ["img1.jpg", "img2.jpg"],"stock": 12,"category": "Electronics","rating": 4.3,"reviews": [{"user": "John", "rating": 5, "comment": "Excellent sound quality!"},{"user": "Priya", "rating": 4, "comment": "Battery life could be better."}]}

Key API Endpoints:

- GET /api/products/{id} → Fetch product details
- POST /api/cart/add → Add product to cart
- POST /api/wishlist/add → Add product to wishlist
- GET /api/reviews/{productId} → Fetch reviews
- POST /api/reviews/{productId} → Add new review

3.Data Handling Approach

To ensure smooth interaction between frontend, backend, and database, the following approach is used:

- Frontend → Uses Axios/React Query for API calls with caching, loading states, and error handling.
- Backend → Validates product ID, stock availability, and price before updating the database.
- Database →
  - products → Stores product details
  - cart → Stores user cart items
  - reviews → Stores customer feedback
- Error Handling →
  - 200 → Success
  - 400 → Invalid Request
  - 404 → Product Not Found
  - 500 → Server Error

4.component/Module diagram

Frontend Components

- ProductPage.js
- ProductImageGallery.js
- ProductInfo.js
- AddToCart.js
- WishlistButton.js
- ReviewsSection.js

Backend Modules

- ProductController.js → Handles product-related APIs
- CartController.js → Manages cart operations
- ReviewController.js → Manages review operations
- AuthMiddleware.js → Handles authentication & authorization

5.Bas ic Flow Diagram

- User visits Product Page
- System calls GET /api/products/{id} → Fetch product details from DB
- UI displays product info, images, price, stock, and reviews
- User clicks Add to Cart → API POST /api/cart/add is triggered
- Backend validates product availability → Updates cart in DB
- API responds with success → Cart count updates on UI
- User can also add product to wishlist or submit reviews