# **COLLEGE CODE** **9627**

# **COLLEGE NAME** **Universal College of Engineering and**

# **Technology**

# **DEPARTMENT** **Computer Science Engineering**

# **ROLL NO** **962723104056**

# **DATE** **05-10-2025**

# **Completed the project named as Phase - 5**

# **TECHNOLOGY PROJECT NAME: Ecommerce Product Page**

**SUBMITTED BY,**

**NAME M. Vidhya**  
**MOBILE NO 6381498626**

# **Ecommerce Product Page**

[GitHub Repo Link: https://github.com/Vidhya2307/Ecommerce-Product-Page.git`](github)

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>Login - Vcart</title>

  <style>

    @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap');

    \* {

      box-sizing: border-box;

      margin: 0;

      padding: 0;

      font-family: 'Poppins', sans-serif;

    }

    body, html {

      height: 100%;

      background: #fafafa;

      display: flex;

      justify-content: center;

      align-items: center;

    }

    .container {

      display: flex;

      width: 900px;

      height: 600px;

      border-radius: 15px;

      box-shadow: 0 18px 40px rgba(0,0,0,0.1);

      overflow: hidden;

      background: white;

    }

    .image-section {

      flex: 1.2;

      background: url('https://static.vecteezy.com/system/resources/previews/003/806/600/non\_2x/smiling-girl-with-shopping-bags-on-sale-in-cartoon-style-vector.jpg') no-repeat center center/cover;

      position: relative;

      color: white;

      padding: 40px;

      display: flex;

      flex-direction: column;

      justify-content: flex-end;

      box-shadow: inset 0 0 0 1000px rgba(0, 0, 0, 0.4);

    }

    .image-section h1 {

      font-size: 58px;

      font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

      font-weight: 700;

      color: #f57224;

      margin-bottom: 15px;

    }

    .image-section h2{

        font-size : 48px;

        color: #faf9f9;

    }

    .image-section p {

      font-size: 18px;

      line-height: 1.4;

      max-width: 280px;

      font-weight: 300;

    }

    .login-section {

      flex: 1;

      padding: 60px 40px;

      display: flex;

      flex-direction: column;

      justify-content: center;

      background: #ffffff;

    }

    .login-section h2 {

      font-size: 32px;

      font-weight: 600;

      color: #222;

      margin-bottom: 40px;

      text-align: center;

      letter-spacing: 1px;

    }

    form {

      display: flex;

      flex-direction: column;

    }

    label {

      font-weight: 600;

      font-size: 14px;

      margin-bottom: 6px;

      color: #555;

    }

    input[type="email"],

    input[type="password"] {

      margin-bottom: 25px;

      padding: 15px 18px;

      border: 1.5px solid #ddd;

      border-radius: 10px;

      font-size: 16px;

      transition: border-color 0.3s;

    }

    input[type="email"]:focus,

    input[type="password"]:focus {

      border-color: #f57224;

      outline: none;

      box-shadow: 0 0 8px rgba(245, 114, 36, 0.4);

    }

    button {

      padding: 15px 0;

      background: #f57224;

      border: none;

      border-radius: 10px;

      color: white;

      font-weight: 600;

      font-size: 18px;

      cursor: pointer;

      transition: background 0.3s;

      letter-spacing: 1px;

    }

    button:hover {

      background: #d05c1c;

    }

    .register-text {

      text-align: center;

      margin-top: 20px;

      color: #777;

      font-size: 14px;

    }

    .register-text a {

      color: #f57224;

      font-weight: 600;

      text-decoration: none;

      transition: text-decoration 0.3s;

    }

    .register-text a:hover {

      text-decoration: underline;

    }

    @media (max-width: 760px) {

      .container {

        flex-direction: column;

        width: 90%;

        height: auto;

      }

      .image-section {

        flex: unset;

        height: 200px;

        padding: 20px;

        box-shadow: none;

      }

      .login-section {

        padding: 30px 20px;

      }

    }

  </style>

</head>

<body>

  <div class="container">

    <div class="image-section">

      <h2>Welcome to </h2><h1>Vcart</h1>

      <p>Experience the best shopping with our exclusive deals and user-friendly platform.</p>

    </div>

    <div class="login-section">

      <h2>Login to Vcart</h2>

      <form action="/login" method="POST" autocomplete="off">

        <label for="email">Email Address</label>

        <input type="email" id="email" name="email" placeholder="you@example.com" required />

        <label for="password">Password</label>

        <input type="password" id="password" name="password" placeholder="Enter your password" required />

        <button type="submit">Login</button>

      </form>

      <p class="register-text">

        Don't have an account? <a href="/register">Register here</a>

      </p>

    </div>

  </div>

</body>

</html>

.App {

  text-align: center;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <img src={logo} className="App-logo" alt="logo" />

        <p>

          Edit <code>src/App.js</code> and save to reload.

        </p>

        <a

          className="App-link"

          href="https://reactjs.org"

          target="\_blank"

          rel="noopener noreferrer"

        >

          Learn React

        </a>

      </header>

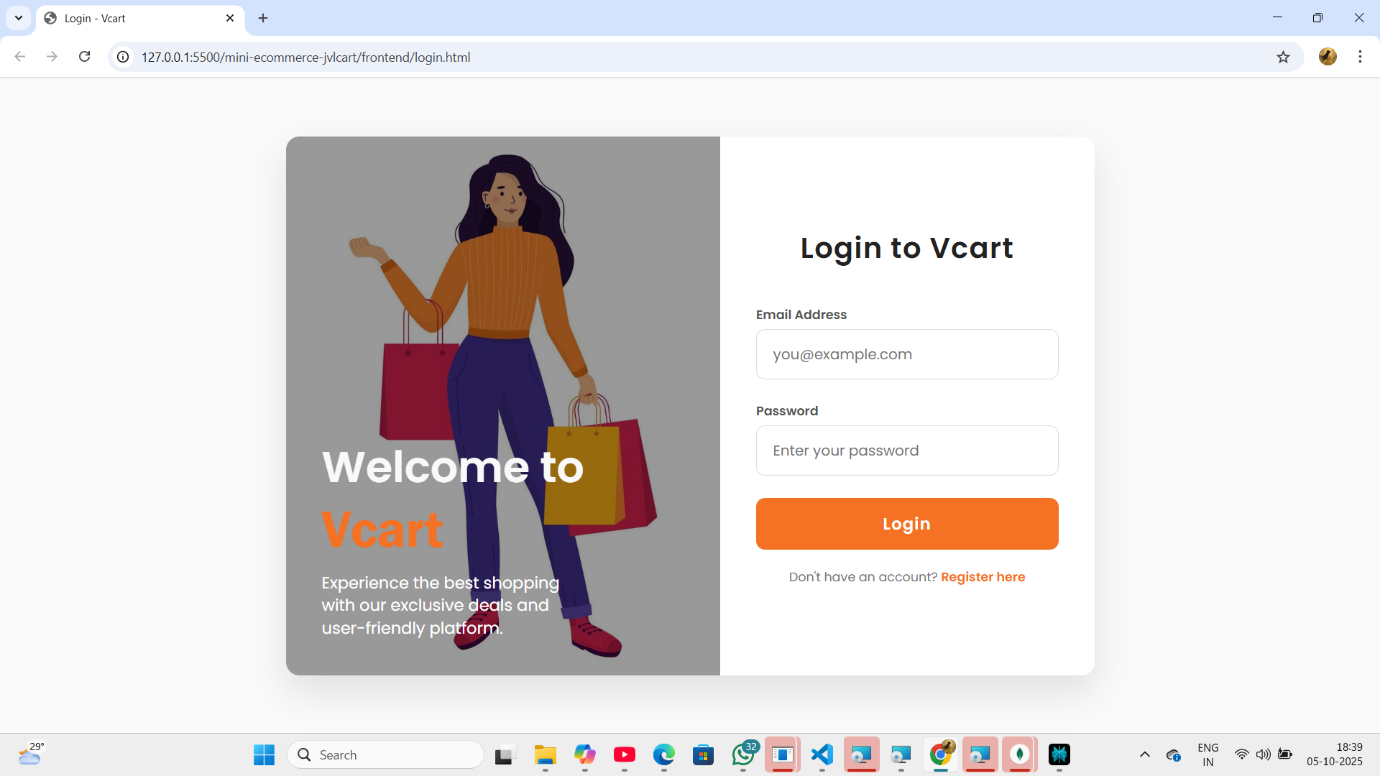
    </div>

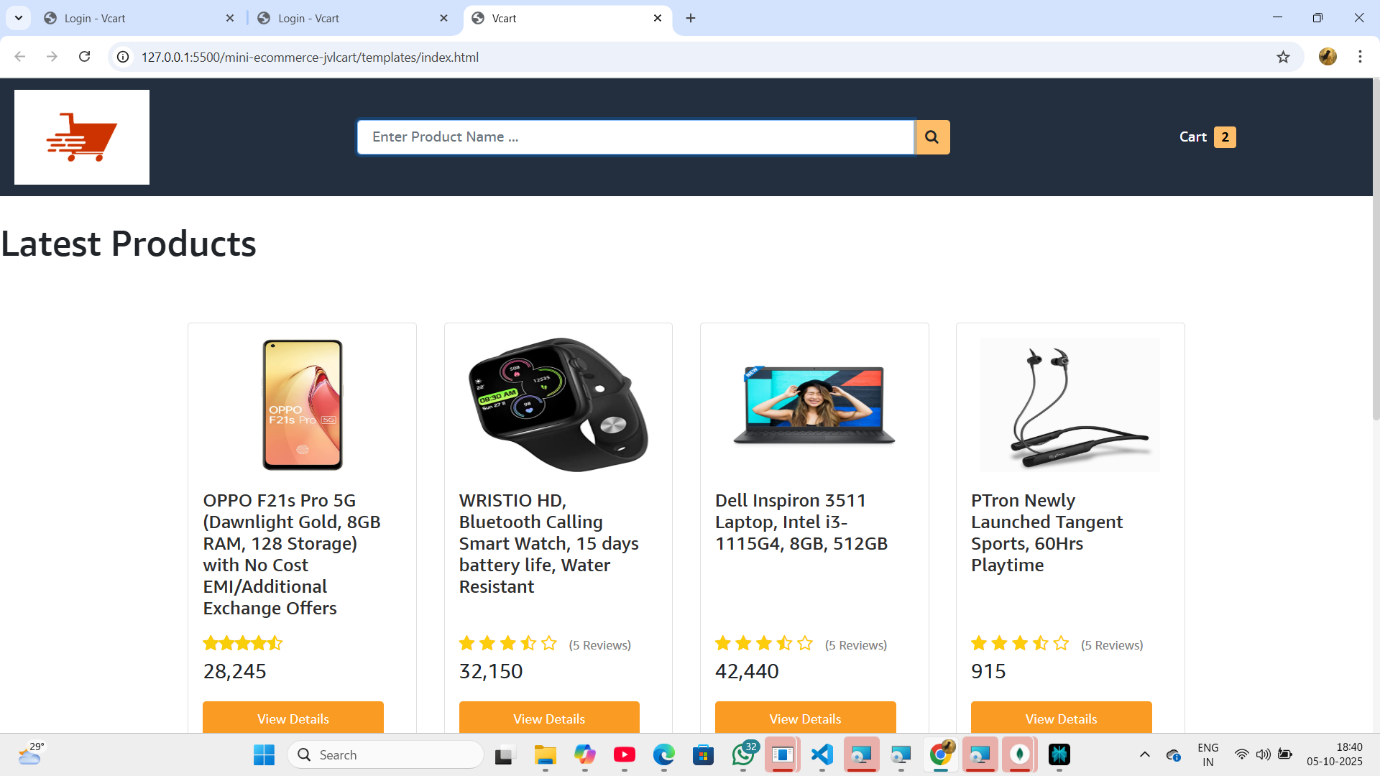
  );

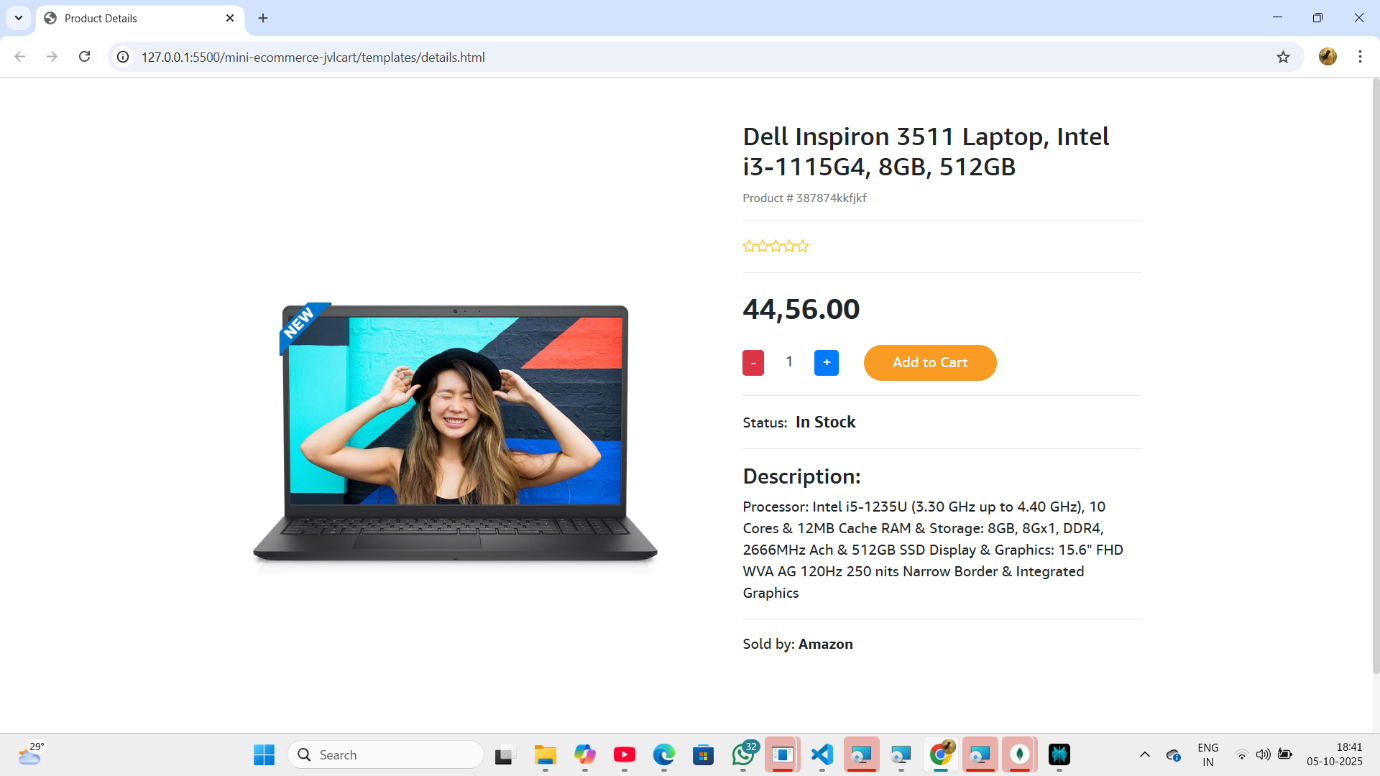
}

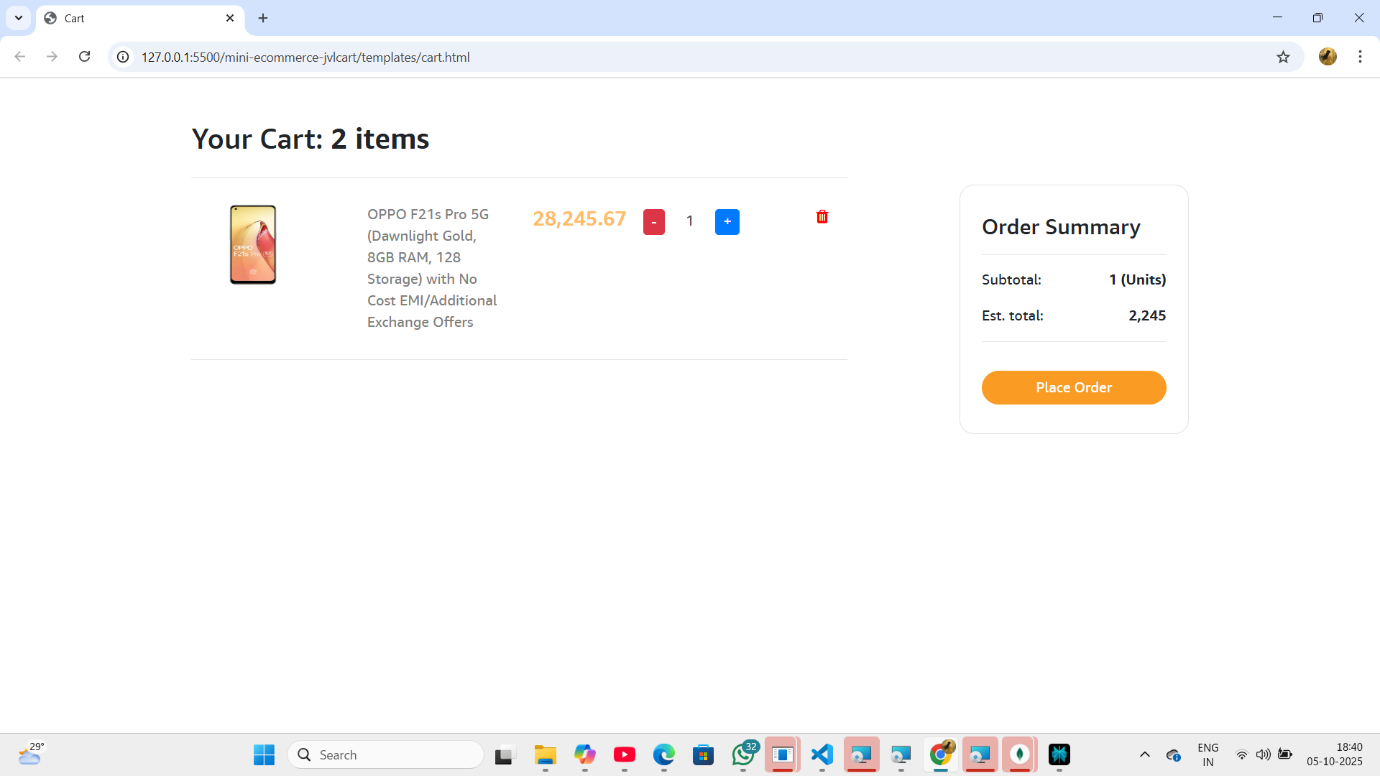
export default App;

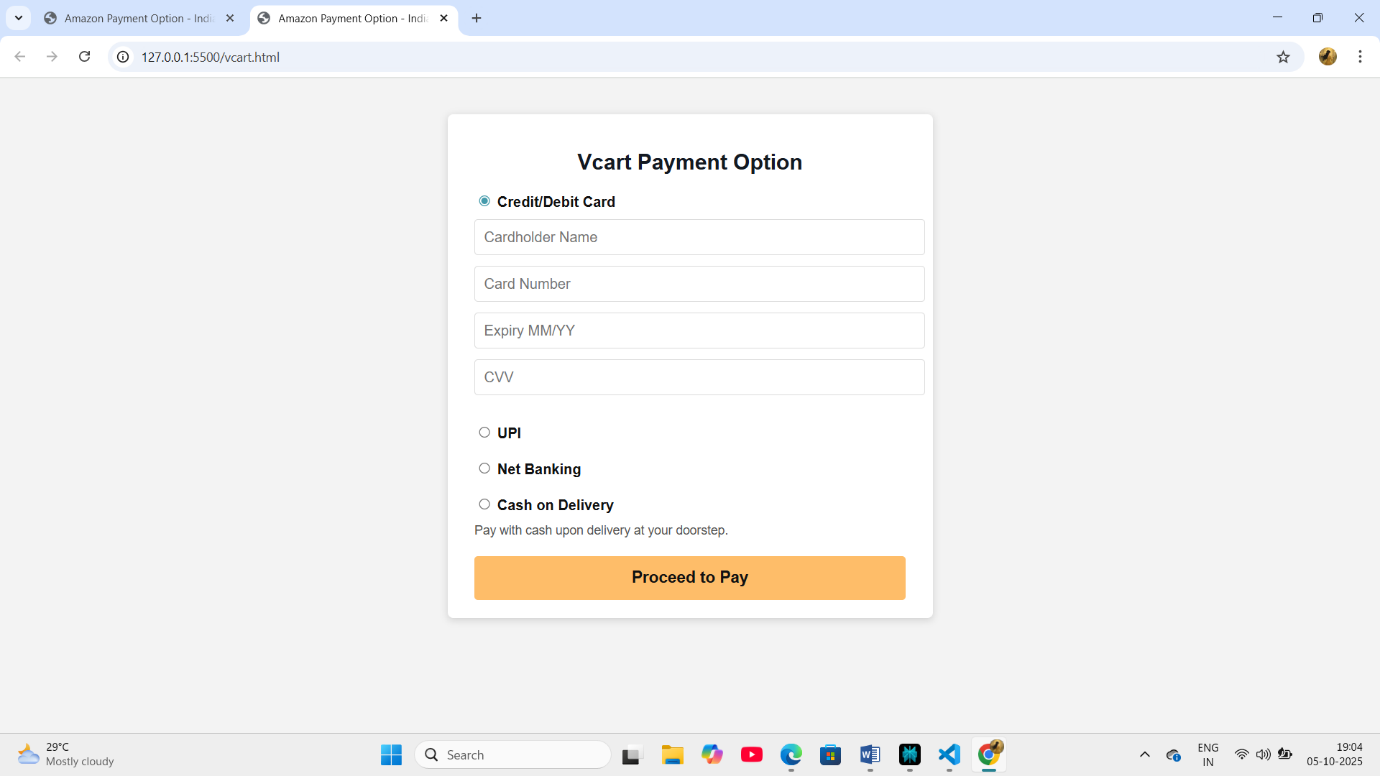
**Screenshots / API Documentation**

****

****

****

****

****

# Project Report

## 1. Abstract

Ecommerce is one of the fastest-growing industries, and a product page forms the backbone of any online shopping experience. This project demonstrates the design and development of a **React-based Ecommerce Product Page** that dynamically displays products, allows users to view product details, and provides a responsive user interface. The project uses **React.js** for building reusable components and **FakeStoreAPI** (or JSON data) as the data source.

The outcome of the project is a **responsive, user-friendly product page** that can be extended into a complete ecommerce application by integrating features like cart, authentication, and payment systems.

## 2. Introduction

The increasing demand for online shopping has led to the development of scalable, interactive, and efficient web applications. A product page is a fundamental part of an ecommerce website where customers browse, filter, and select products.

## 3. Objectives

The main objectives of this project are:

1. To design and implement a **React-based ecommerce product page**.
2. To fetch and display products dynamically using an **API/JSON data source**.
3. To provide a **responsive design** for seamless user experience on mobile, tablet, and desktop.
4. To implement **modular components** for better scalability and maintainability.
5. To explore the possibility of extending the project into a **full-fledged ecommerce platform**.

## 4. Literature Review

* **Traditional Web Development:** Earlier ecommerce sites were built using static HTML, CSS, and JavaScript, making maintenance difficult.
* **Modern Frontend Frameworks:** With frameworks like **React** building interactive UI has become easier and efficient.
* **Ecommerce Trends:** Studies show that **70% of customers leave** a shopping site due to poor UI/UX. Thus, a responsive and interactive product page is crucial.
* **Existing Solutions:** Websites like Amazon and Flipkart heavily rely on **dynamic product rendering and modular design**, which inspired this project.

## **5.System Design Notes: React Ecommerce App**

### 5.1. **Frontend (React)**

* **Tech Stack**: React, React Router, Axios, Redux (optional)
* **Responsibilities**:
  + Product listing and filtering
  + Cart and checkout flow
  + User authentication (login/signup)
  + Order history and user profile
  + Responsive UI with reusable components

### 5.2. **Backend (Node.js + Express)**

* **Tech Stack**: Node.js, Express, MongoDB (or PostgreSQL), JWT
* **Responsibilities**:
  + RESTful APIs for products, users, orders
  + Authentication and authorization
  + Cart and order management
  + Admin panel for product CRUD
  + Payment gateway integration (e.g., Razorpay, Stripe)

### 5.3. **Database Design**

| **Collection/Table** | **Fields** |
| --- | --- |
| Users | name, email, password, role, address |
| Products | name, description, price, stock, category |
| Orders | userId, items, total, status, paymentInfo |
| Cart (optional) | userId, productId, quantity |
| Reviews | userId, productId, rating, comment |

### 5.4. **Authentication**

* JWT-based login system
* Role-based access (Admin vs Customer)

### 5.5. **Admin Features**

* Add/edit/delete products
* View all orders
* Manage users

## 🔄 Workflow Notes: Project Setup & Development

### ✅ Step 1: Fix create-react-app Error

npm install -g create-react-app

create-react-app react-ecommerce-app

### ✅ Step 2: Project Initialization

* Create two folders:
  + ecommerce-frontend → React app
  + ecommerce-backend → Node.js/Express server

### ✅ Step 3: Backend Setup

npm init -y

npm install express mongoose cors dotenv bcryptjs jsonwebtoken

* Create folders: models, routes, controllers
* Setup MongoDB connection
* Build APIs: /api/products, /api/users, /api/orders

### ✅ Step 4: Frontend Setup

npx create-react-app ecommerce-frontend

cd ecommerce-frontend

npm install axios react-router-dom

* Create pages: Home, Product, Cart, Login, Signup
* Use Axios to fetch data from backend
* Implement protected routes

### ✅ Step 5: Testing & Debugging

* Use Postman for backend API testing
* Use browser dev tools for frontend debugging
* Resolve npm errors by clearing cache:
* npm cache clean --force

### ✅ Step 6: Deployment

* Backend: Render, Railway, or Heroku
* Frontend: Vercel or Netlify
* Use .env files for secrets and API URLs

## 6. Implementation

### 6.1 Technologies Used

* **React.js** → Component-based UI framework
* **CSS**  → Styling and responsiveness
* **JavaScript**  → Logic and interactivity
* **JSON** → Product data

### 6.2 Key Features Implemented

* Product list rendering using map()
* Product detail display using props and state
* API data fetching with useEffect and fetch()
* Responsive design with CSS grid/flexbox

## 7. Results

* Successfully displayed products dynamically.
* Provided detailed product view on click.
* Ensured mobile responsiveness.
* Code structured into **reusable components**, enabling scalability.

## 8. Conclusion

This project successfully demonstrates the implementation of an **Ecommerce Product Page using React**. It provided hands-on experience in **React components, hooks, API integration, and responsive design**.

The project lays the foundation for a **scalable ecommerce application**. With additional features like cart management, authentication, and payment integration, it can evolve into a fully functional ecommerce website.

## 9. Future Scope

* Add **Add to Cart & Checkout** functionality.
* Implement **Search, Filter, and Sort** options.
* Add **User Authentication** using Firebase/Node.js.
* Integrate a **real backend and database**.
* Enable **Payment Gateway Integration** for real transactions.

## **Challenges & Solutions**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Challenge** | **Solution** | | Understanding React component lifecycle | Learned useEffect hook for API calls | | Passing data between components | Used React **props** and **state management** | | Styling consistency | Used **CSS Grid/Flexbox** and Tailwind classes | | Rendering product details dynamically | Used **conditional rendering** and useState | | |  | | --- | | Deployment errors |  |  | | --- | |  | | Used **Netlify** for smooth deployment | |  |

## GitHub README & Setup Guide

**Features**

Dynamic product listing

Product details view

Responsive design (mobile/tablet/desktop)

**Tech Stack**

React.js

CSS

FakeStoreAPI

**Setup Instructions**

1. Clone the repository

git clone https://github.com/your-username/ecommerce-product-page.git

1. Install dependencies
2. npm install
3. Run the project
4. npm start

6. Final Submission