

**COLLEGE CODE: 9504**

**COLLEGE NAME: DR.G.U.POPE COLLEGE OF ENGINEERING**

**DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING**

**STUDENT NM ID: 63590E8575E1AEC83C4481F08DFA50DA**

**ROLL NO: 950423104044**

**DATE: 06-10-2025**

**Completed the project SINGLE PAGE APPLICATION Phase-05**

**TECHNOLOGY PROJECT NAME: IBM-FE- SINGLE PAGE APPLICATION**

**SUBMITTED BY,**

**NAME: Suresh Kumar R**

**Mobile: 9524918976**

## Phase 5 — Project Demonstration & Documentation

### Final Demo Walkthrough:

- The final demonstration showcases the complete working of the E-commerce Clothing Site built as a Single Page Application (SPA).
- The demo begins with the Home Page, highlighting featured clothing collections and offers.
- Users can browse products through categorized sections like Men, Women, and Kids.
- The Product Detail Page displays images, price, and size options dynamically fetched via API.
- The Add to Cart and Checkout flows are demonstrated to show smooth navigation without page reloads.
- The Admin Dashboard (if included) showcases product management and analytics features.
- The UI is responsive, fast, and provides a seamless shopping experience.

### Project Report:

#### The report includes:

**Introduction** – Explains the objective of developing a modern, user-friendly e-commerce SPA for clothing.

**System Design** – Depicts the frontend-backend architecture and data flow.

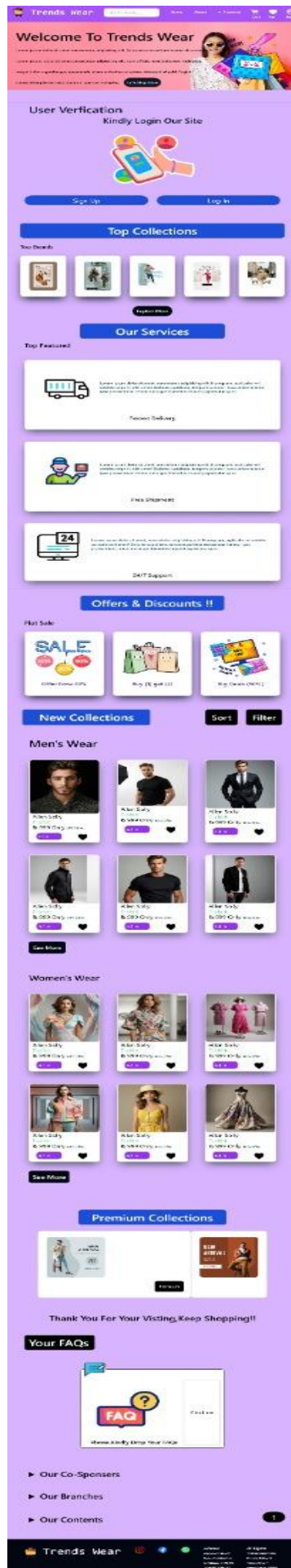
**Modules** – Lists core modules: User Authentication, Product Listing, Cart Management, Payment Simulation, and Admin Controls.

**Testing & Deployment** – Describes test cases, bug fixing, and final deployment on a hosting platform (like Netlify or Vercel).

**Conclusion** – Summarizes learning outcomes and future enhancements.

### Screenshots:

Screenshots cover all major screens: Home, Product Listing, Product Details, Cart, Checkout, and Admin Panel.



## Challenges & Solutions:

Challenge	Solution
Maintaining fast navigation without reloading pages Router for client-side routing	Implemented React
Managing global cart state across components for centralized state management	Used Context API / Redux
Handling dynamic product data and caching logic	Integrated API layer with proper fetch
UI responsiveness on all devices queries	Applied CSS grid/flexbox with media
Deployment issues with API routes variables properly	Configured proxy & environment

## GitHub README & Setup Guide:

The GitHub repository includes a detailed README.md file containing:

Project Overview & Features

Tech Stack Used: React.js, Node.js (optional backend), CSS/Tailwind, JSON or API data

Installation Steps:

```
git clone https://github.com/yourusername/ecommerce-spa.git
```

```
cd ecommerce-spa
```

```
npm install
```

```
npm start
```

Deployment Guide: Steps for hosting on Netlify or Vercel.

Credits & License section.

## **Final Submission (Repo + Deployed Link):**

**GitHub Repository:** [https://github.com/Suresh-Sivathai/Naan\\_mudhalvan\\_project-SPA-.git](https://github.com/Suresh-Sivathai/Naan_mudhalvan_project-SPA-.git)

**Deployed Link:** [https://reniston-r.github.io/Single\\_page\\_application/](https://reniston-r.github.io/Single_page_application/)

**Both links are submitted for final evaluation, including:**

**Working application**

**Source code with comments**

**Proper documentation and screenshots**