Project Documentation: CAN – AI-Powered E-Commerce Platform

---

1. Introduction

Overview of the Project

CAN is a feature-rich, AI-powered e-commerce platform developed for a modern clothing brand aiming to redefine the digital shopping experience. Built using cutting-edge web technologies, the platform allows users to browse, personalize, and purchase fashion items seamlessly. One of its standout features is an AI stylist that recommends products based on user preferences and trends. An intuitive admin dashboard enables effortless store management, inventory control, and content publishing.

---

2. Background of the Project

Problem Statement and Motivation

Traditional e-commerce platforms often fall short in delivering truly personalized shopping experiences, especially in the fashion domain. Shoppers find it difficult to discover items that match their style, and business owners struggle with managing large catalogs, promotions, and customer engagement.

CAN was born from the motivation to solve these issues by:

- Empowering users with AI-driven fashion recommendations.
- Simplifying store operations for administrators.
- Delivering a smooth, modern UI/UX for online fashion retail.

---

3. Objectives

- Build a scalable and user-friendly e-commerce solution tailored to fashion retail.
- Integrate AI to deliver real-time, personalized style suggestions.
- Provide complete store control through a secure, well-designed admin dashboard.
- Offer essential features: user accounts, cart, wishlist, checkout, and order tracking.

- Use best practices in full-stack development for security, maintainability, and performance.

---

4. Scope

Features Included:

- User authentication (signup, login, password reset)
- Product catalog with category-wise filtering and search
- Shopping cart, order placement, and user dashboard
- Wishlist management and address book
- Admin panel for product, category, order, user, and promotion management
- AI-powered stylist for personalized recommendations
- Blog and content management system

Features Not Included

- Real payment processing (demo transactions only)
- Mobile application (web version only)
- Multi-language and multi-currency support
- Deep analytics and BI dashboards (future scope)

---

5. Literature Review / Related Work

Existing Systems:

- Amazon, ASOS, Zalando: Industry leaders using advanced recommendation engines.
- Shopify, WooCommerce: Widely used for e-commerce businesses but lack deep AI integration.
- Stitch Fix: Specialized in AI-powered fashion personalization.

Relevant Models & Theories:

- **Collaborative Filtering & Content-Based Filtering:** Core AI techniques for recommendations.

- **MVC (Model-View-Controller) Architecture:** Organizes code for maintainability and clarity.
- **Agile Development Methodology:** Adopted for iterative, test-driven development.

---

6. Methodology

Technologies Used:

- **Frontend:** Next.js (React, TypeScript), Tailwind CSS
- **Backend:** Next.js API Routes (Node.js)
- **Database:** MongoDB
- **Version Control:** GitHub
- **Styling Tools:** Post CSS, Tailwind Plugins
- **AI Used :** ChatGPT, Gemini AI, Grok AI etc.

Development Phases:

1. Requirement Analysis & Planning
2. UI/UX Wireframing and Prototyping
3. Backend and Database Configuration
4. Component-based Frontend Development
5. AI Integration and Optimization
6. Testing, Debugging, and QA

Design Artifacts:

- ER Diagrams: Class diagrams for Users, Products, Orders.
- Flowcharts: Checkout process, AI recommendation logic, Admin workflow.

---

7. Implementation

Development Summary:

- Initiated a scalable project architecture using Next.js and TypeScript.
- Designed a responsive UI with reusable components (Navbar, Modals, Forms).
- Developed user authentication and session management.

- Implemented product listing, detail views, cart logic, and checkout flows.
- Built an admin dashboard with CRUD operations for all resources.
- Integrated the AI Stylist to recommend outfits based on user interactions.
- Connected the app to MongoDB for persistent data handling.

Sample Code Snippet – Fetching Products from MongoDB:

```typescript
// src/lib/services/productService.ts
export async function getProducts() {
  return await db.collection('products').find({}).toArray();
}
```

System Architecture Overview:

```scss
User Interface (Next.js + Tailwind)
         ↓
API Routes (Node.js Backend)
         ↓
Recommendation Engine (AI Module)
         ↓
Database (MongoDB)
```

Database Schema Snapshot:

- Users: `{ _id, name, email, password, addresses[], wishlist[] }`
- Products: `{ _id, name, description, price, images[], category }`
- Orders: `{ _id, userId, items[], status, total }`
- Categories, Coupons, BlogPosts, etc.`

---

8. Results & Achievements

- Developed a fully functional, responsive e-commerce clothing platform.
- Enabled real-time AI-powered fashion suggestions for enhanced UX.
- Delivered a powerful admin panel to manage all store components.

- Ensured secure and structured backend integration.
- Maintained modular architecture, suitable for scaling and feature expansion.

---

9. Challenges Faced & Resolutions

Challenge	Resolution
Integrating AI with dynamic user data	Created a service layer for fetching and preprocessing user behavior data
Authentication and secure backend routes	Used encrypted tokens and session-based auth
State management in complex UI	Adopted context-based state and efficient API caching
Designing an intuitive yet powerful admin UI	Used layout scaffolding with clear user flow diagrams

---

10. Future Scope

- Integrate real payment gateways (Bkash, PayPal)
- Launch a mobile app using React Native
- Add multi-language and multi-currency support
- Build advanced AI models for style prediction and trend analysis
- Introduce analytics dashboards for admin insight and growth tracking
- Incorporate chatbot support for shopping assistance

11. Key Features Highlight

Include a visually engaging summary of standout features:

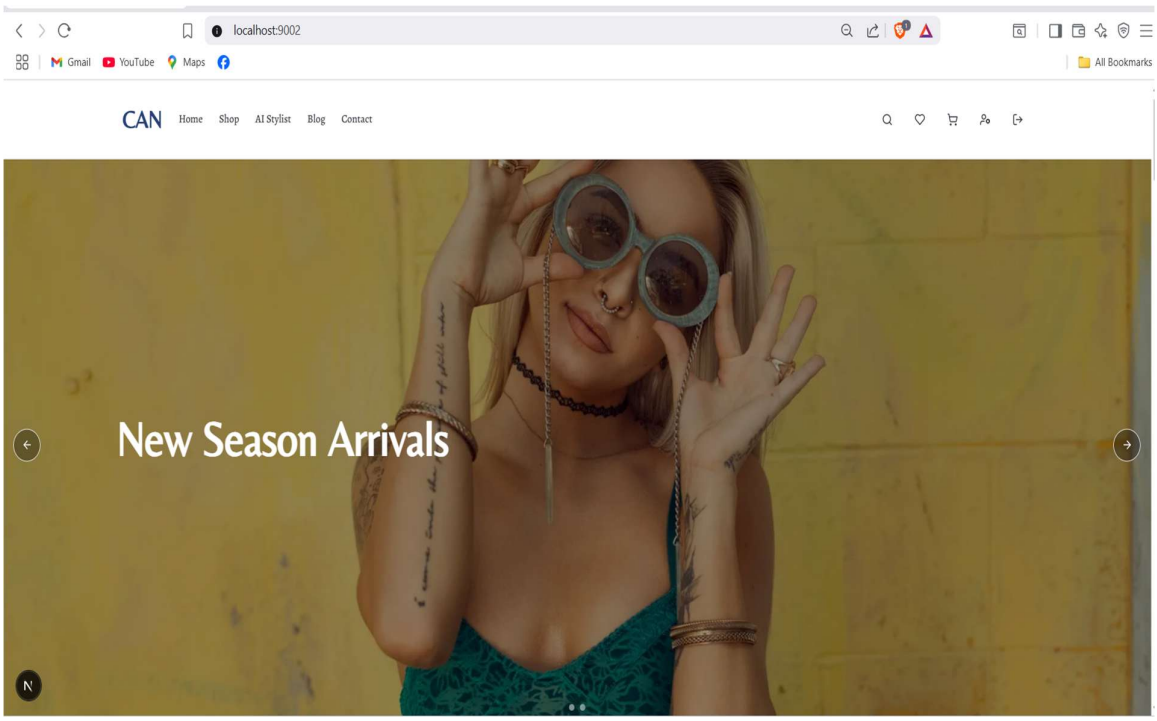
Feature	**Description**
AI Stylist	Personalized outfit recommendations based on user preferences and history.
One-Click Checkout	Streamlined cart-to-order process.

Feature	Description
Order Tracking	Real-time status updates on order lifecycle.
Dashboard Analytics (Basic)	Visual insights on user activity and sales (even simple graphs).
Blog Integration	Content marketing support through CMS.
JWT Auth	Secure login and token-based route protection.

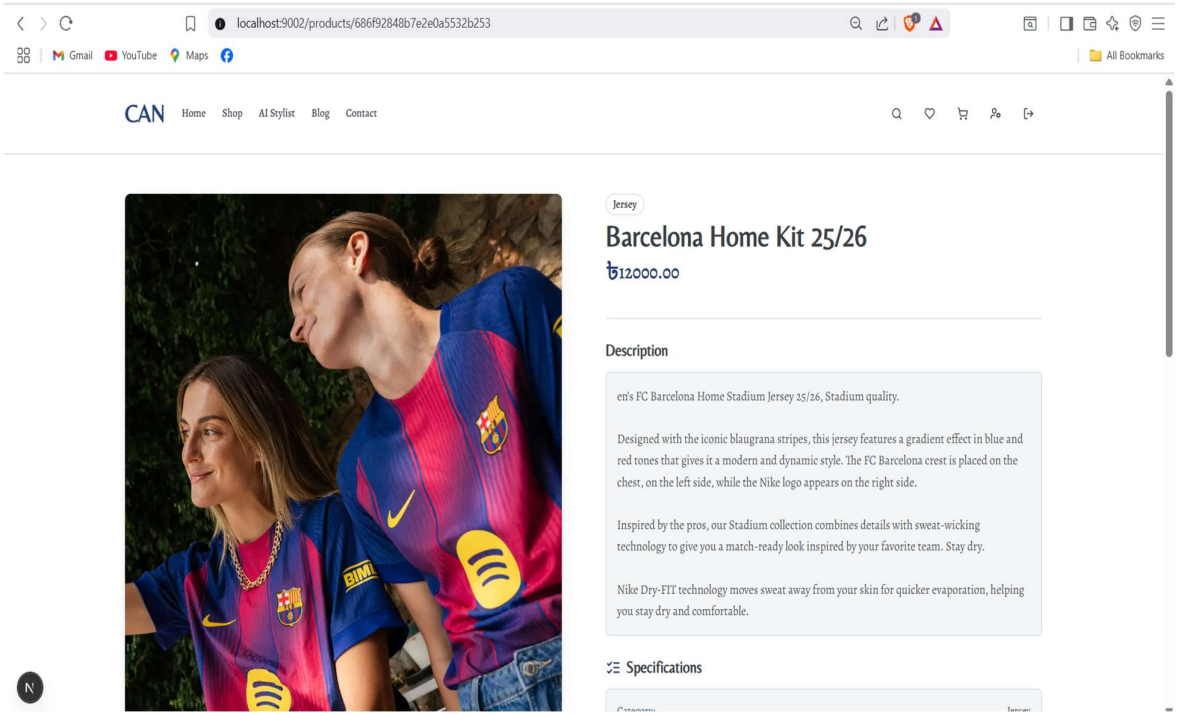
---

12. UI Snapshots / Screens

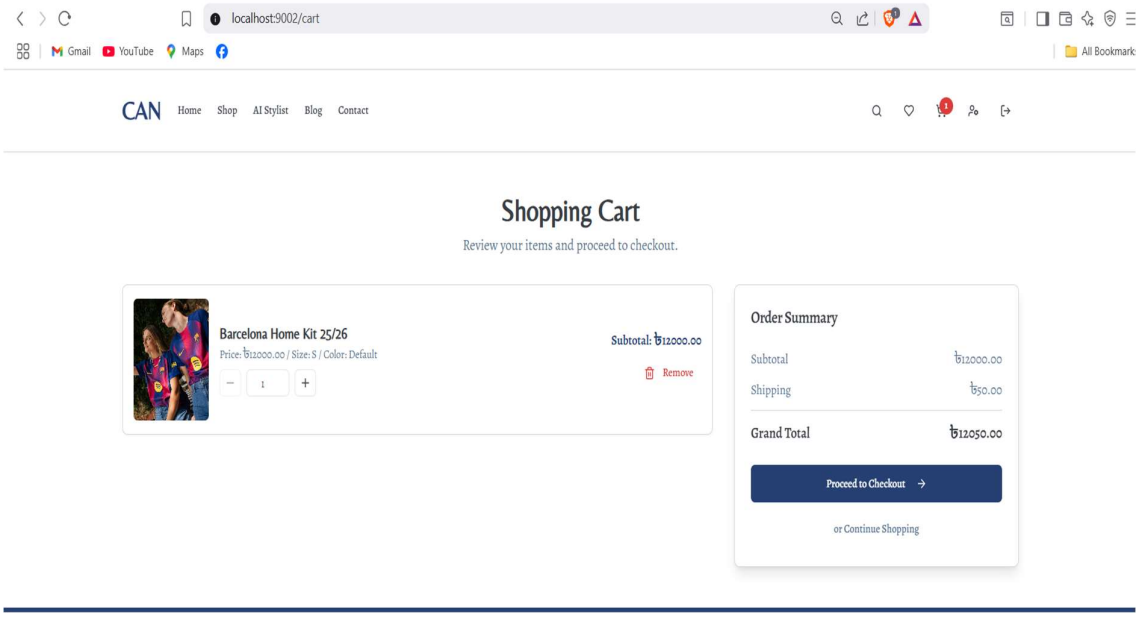
Add annotated screenshots for:

- **Home Page**
- 

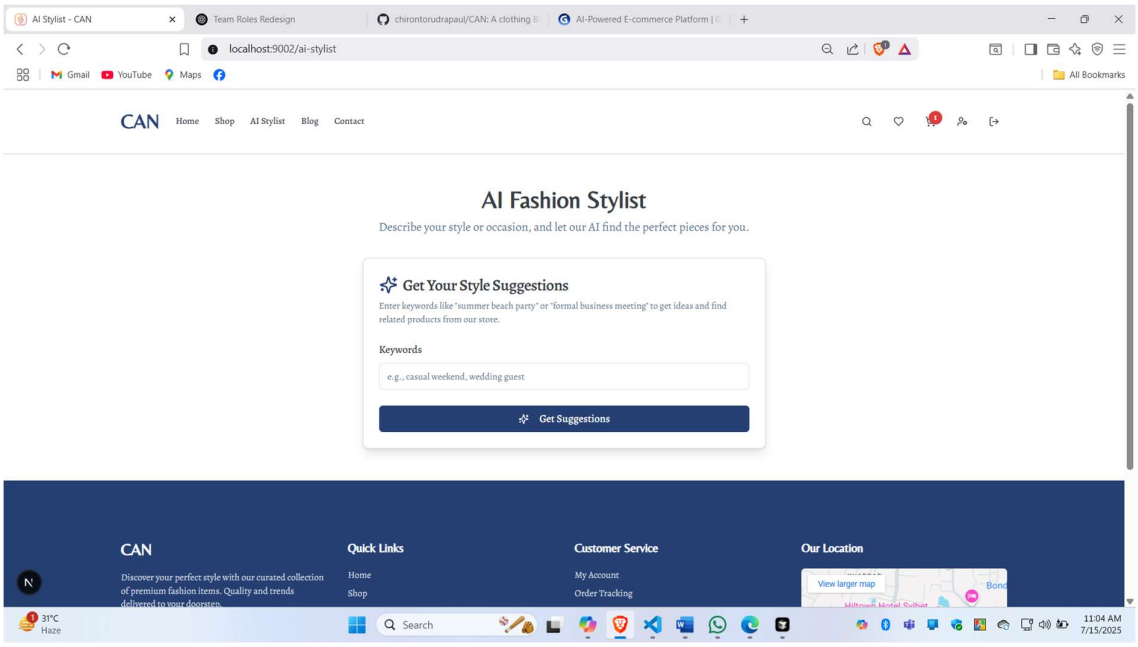
- **Product Detail Page**



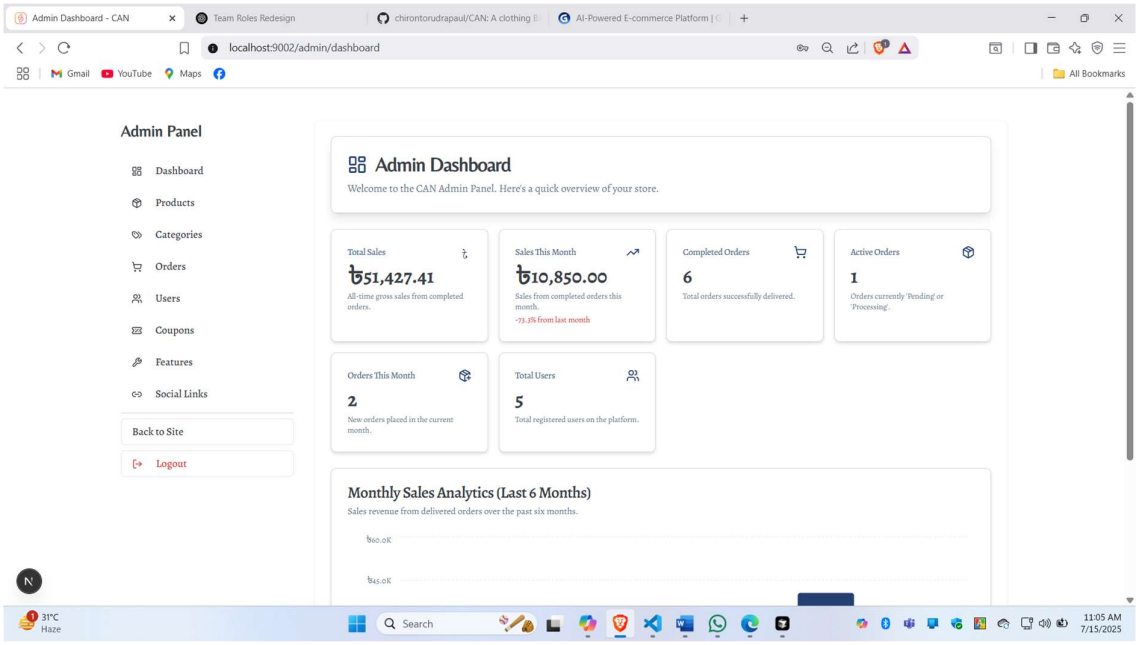
- **Cart & Checkout**



- **AI Stylist Interface**



- **Admin Dashboard**



---

13. Performance Optimizations

Demonstrate professionalism by showing awareness of speed and efficiency:

- Lazy loading product images
- Static site generation (SSG) or server-side rendering (SSR) with Next.js
- Indexed MongoDB queries
- Pagination and infinite scroll
- Reduced bundle size using dynamic imports

---

14. Testing Strategy

Briefly outline how the app was tested:

- **Unit Tests** – Product and Auth services using Jest
- **Integration Tests** – Cart and checkout flows
- **UI Testing** – With Playwright or Cypress (if applicable)
- **Manual QA** – On Chrome, Firefox, and Microsoft Edge.

---

15. Team & Roles

List team members and contributions:

Member	**Role**	**Contribution**
Najir	**Full Stack Developer**	**UI design, Frontend & Backend for customer features**
Chironto	**Admin Panel**	**Frontend & Backend development.**
Arnob	**Database & Testing**	**Database management, System testing.**

---

16. User Manual / Walkthrough Guide (Quick Steps)

1. Sign Up

- Go to homepage → Click Sign Up
- Fill in name, email, password → Click Create Account
- Verify email → Log in

2. Use AI Stylist

- Log in → Go to AI Stylist

- Type style/occasion
- Click Get Recommendations
- View suggestions → Add to cart/wishlist

3. Add/Edit Products (Admin)

- Log in as Admin → Open Admin Dashboard
- Add Product: Fill form → Upload image → Save
- Edit Product: Click edit icon → Update → Save
- Delete Product: Click delete icon → Confirm

---

17. Version Control & Git Strategy

Mention:

- GitHub repo link : https://github.com/chirontorudrapaul/CAN_v5
- Branching strategy used : main branch

18. Conclusion

CAN is more than an online clothing store—it's a smart, scalable, and modern e-commerce ecosystem. It combines the power of artificial intelligence with a clean user experience to enhance fashion retail. Both customers and administrators benefit from streamlined workflows, intelligent features, and a thoughtfully designed platform.

Project Contributors

This project was collaboratively developed by a dedicated student team as part of the Software Engineering / Final Year Project.

Name	Role	Institution	ID
Najir Hossain Shahinur	Full Stack Developer	North East University Bangladesh, Sylhet	0562310005101033
Chironto Rudra Paul	UI/UX Designer & Frontend Assistant	North East University Bangladesh, Sylhet	0562310005101048
Arnob Das	Database & Testing	North East University Bangladesh, Sylhet	0562310005101028

Project Title: CAN – AI-Powered E-Commerce Platform for Fashion Retail
Course: Software Engineering and Development Paradigm
Course code : CSE-06133114

Supervisor: Sabuj Chandra Paul
Department: Computer Science & Engineering (CSE)
Session: Spring - 2025