E-Commerce Shopping Website Documentation

Introduction

This documentation provides an in-depth overview of the E-Commerce Shopping Website, outlining its purpose, features, technologies, and setup instructions. The platform enables users to browse, select, and purchase clothing items with a seamless shopping experience.

Features

1. User Authentication

- Sign up and login functionality using Clerk authentication.
- Secure user session management.

2. Product Categories

- Wide range of clothing items including:
    - Shoes
    - Tops
    - T-Shirts
    - Hoodies
    - Jackets
    - Trousers
    - Tights
    - Shorts
    - Tracksuits
    - Jumpsuits & Rompers
    - Skirts & Dresses
    - Socks & Accessories
    - Casual Wear
    - Formal Shirts
    - Denim & Jeans
    - Polo T-Shirts
    - Blazers
    - Yoga Pants
    - Winter & Summer Collections (Men, Women, Kids)

3. Dynamic Product Listings

- Fetching product data using Sanity API.
- Dynamic routing for each product category and individual product pages in Next.js.

4. Shopping Cart Functionality

- Add and remove products from the cart.
- Display product details including name, price, and quantity.
- Cart updates without using any state management library.

5. Secure Checkout & Payment Integration

- Stripe payment gateway integration for secure transactions.
- Responsive and user-friendly checkout process.

6. Shipment Integration

- Integration with ShipEngine for efficient shipping and tracking.
- Automatic generation of shipping labels.
- Real-time shipment tracking updates.

7. Responsive Design

- Fully optimized UI for all screen sizes using Tailwind CSS.
- Mobile-first design approach.

8. Optimized Performance

- Server-side rendering (SSR) and static site generation (SSG) for enhanced performance.
- Optimized images and assets for fast loading.

Technologies Used

- **Framework:** Next.js (React-based framework)
- **Styling:** Tailwind CSS for efficient and responsive styling
- **Authentication:** Clerk for user authentication
- **Database & CMS:** Sanity for dynamic content management
- **API Calls:** Fetching data from Sanity API
- **State Management:** Local state handling without external libraries
- **Payment Integration:** Stripe for secure transactions
- **Shipment Integration:** ShipEngine for handling shipping and tracking

Installation & Setup

Prerequisites

Ensure you have the following installed:

- Node.js (Latest LTS version)
- npm or yarn
- A Sanity project with necessary datasets
- Stripe account for payment processing
- ShipEngine API key for shipment management

Steps to Install

1. **Clone the repository**

   ```
   sh
   CopyEdit
   git clone https://github.com/your-repo/ecommerce.git
   ```

2. **Navigate to the project directory**

   ```
   sh
   CopyEdit
   cd ecommerce
   ```

3. **Install dependencies**

   ```
   sh
   CopyEdit
   npm install  # or yarn install
   ```

4. **Set up environment variables**
   - Create a `.env.local` file and add the necessary keys:

     ```
     sh
     CopyEdit
     NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
     NEXT_PUBLIC_SANITY_DATASET=your_dataset
     NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=your_clerk_key
     STRIPE_SECRET_KEY=your_stripe_key
     SHIPENGINE_API_KEY=your_shipengine_key
     ```

5. **Run the development server**

   ```
   sh
   CopyEdit
   npm run dev  # or yarn dev
   ```

6. **Access the site**
   - Open `http://localhost:3000/` in your browser.

Deployment

The project can be deployed using Vercel for optimal performance:

1. **Push the project to GitHub**
2. **Connect the repository to Vercel**
3. **Set up environment variables on Vercel**
4. **Deploy the project**