**DATA DEFENDERS FASHION E-COMMERCE WEBSITE DOCUMENTATION**

**Overview**

Welcome to Data Defenders Fashion e-Commerce website! This platform aims to connect and promotes the rich, vibrant styles of Nigerian fashion industry by providing a marketplace where customers can shop for locally crafted clothing, for any kind of weather and for any gender.

Our mission is to make Data Defenders fashion wears accessible globally and connect customers with authentic, high-quality products from talented Nigerian designers and sellers.

**Table of Contents**

1.About Data Defenders Fashion House

2.Key features

3.Technologies used

4.User instructions

5.Contributions

6.Mission

7.License

**Key Features**

- User-Friendly Interface: Intuitive design for easy browsing and checkout.

- Product Categories: Organized sections for traditional wear, contemporary fashion, accessories, and footwear.

- Secure Payments: Integration with popular Nigerian payment gateways (e.g., Paystack, Flutterwave).

- Customer Account Management: Users can create accounts, view order history, and track orders.

- Shopping Cart & Wishlist: Allows customers to save products for later or add them to their cart.

- Multi-Language and Currency Support: Localized in English and supports Nigerian Naira and USD.

- Product Ratings & Reviews: Customers can rate products and share reviews.

- Admin Dashboard: Manage products, orders, and customer queries efficiently.

---

Technologies used.

Prerequisites

- Node.js: Ensure Node.js is installed (v14+ recommended)

- MongoDB: Set up MongoDB for local or cloud-based data storage

- Git: For version control and cloning the repository

Steps

1. Clone the Repository:

      git clone https://github.com/yourusername/datadefenders-fashion-ecommerce.git

   cd datadefenders-fashion-ecommerce

2. Install Dependencies:

      npm install

3. Set up Environment Variables:

   Create a .env file in the project root and add the following:

      PORT=3000

   MONGODB\_URI=your\_mongodb\_connection\_string

   JWT\_SECRET=your\_jwt\_secret

   PAYMENT\_SECRET\_KEY=your\_payment\_gateway\_secret\_key

4. Start the Development Server:

      npm run dev

   The application should now be running at http://localhost:3000.

User instructions

1. Visit the Homepage: Browse through categories, featured products, New arrivals,crazy deals, summer/spring offers.

2. Shop: Create an account to save your shopping preferences.

3. Add to Cart: Select products and add them to the cart for purchase.

4. Checkout: Complete the purchase using secure payment gateways.

5. Order Tracking: Track your orders in the "My Orders" section.

Technologies Used

- Frontend: React.js, Tailwind CSS, and Redux for state management

- Backend: Node.js, Express.js, and MongoDB

- Payment Integration: Visa or MasterCard or Maestro

- Deployment: AWS for server hosting, Cloudflare for CDN

---

## Project Structure

├── public

├── src

│   ├── components          # Reusable UI components

│   ├── pages               # Application pages (e.g., Home, Product, Cart)

│   ├── services            # API service calls

│   ├── utils               # Helper functions and utilities

│   └── App.js              # Main app component

├── server

│   ├── controllers         # Backend logic (e.g., products, users, orders)

│   ├── models              # Mongoose models

│   ├── routes              # API routes

│   └── server.js           # Express server setup

└── .env                    # Environment variables

## API Documentation

### User Authentication

- POST /api/auth/register: Registers a new user

- POST /api/auth/login: Authenticates a user and returns a JWT token

### Products

- GET /api/products: Fetch all products

- GET /api/products/:id: Fetch product details by ID

- POST /api/products: (Admin) Add a new product

### Orders

- POST /api/orders: Create a new order

- GET /api/orders/:userId: Get all orders for a specific user

For full API documentation, please refer to the [API Documentation File](docs/api.md).

Contributing

Contributions are welcome! Follow these steps to contribute: