

Overall Project Documentation - Clothing E-Commerce

1. Executive Summary

This project is a full-stack e-commerce application designed for clothing retail. It features a modern React frontend, a scalable Node.js/Express backend, and a flexible MongoDB database. The system is designed for high performance, ease of use, and detailed business analytics.

2. System Architecture

The application follows a decoupled client-server architecture: - **Client Side:** Single Page Application (SPA) built with React. - **Server Side:** RESTful API built with Express.js. - **Database:** NoSQL MongoDB for document storage.

Data Flow Diagram (High Level)

- User Interaction:** Frontend (React) sends HTTP requests via Axios.
- Processing:** Backend (Express) validates requests, processes logic, and communicates with the Database.
- Data Persistence:** MongoDB stores user, product, and order data.
- Response:** Backend returns JSON data to the Frontend for rendering.

3. Technology Stack Summary

Layer	Technology	Key Use Case
Frontend	React, Vite, Axios	User Interface & State Management
Backend	Express.js, Node.js	Business Logic & API Routing
Database	MongoDB, Mongoose	Data Storage & Schema Modeling
DevOps	Docker, Docker-compose	Containerization & Local Development
Security	JWT, Bcrypt	Auth & Data Protection

4. Key Implementation Highlights

- **Microservice-like DB Management:** A dedicated `db-manager` container handles database lifecycle tasks independent of the main API.
- **Analytics System:** Built-in tracking for user behavior and product popularities to drive business insights.
- **Responsive Design:** Optimized for various devices using modern CSS practices.

5. Deployment Recommendation

The project is containerized using Docker, allowing for easy deployment to cloud providers like AWS, GCP, or Azure using services like ECS, GKE, or App Service with minimal configuration changes.