

## 1. Overview

---

The backend is a robust RESTful API built with **Node.js** and **Express**, responsible for business logic, authentication, and data management.

## 2. Technology Stack

---

- **Runtime:** Node.js
- **Framework:** Express.js 5.2.1
- **Database ORM:** Mongoose (MongoDB)
- **Authentication:** JSON Web Tokens (JWT) & Bcrypt for password hashing.
- **Email Service:** Nodemailer (for order confirmations/notifications).
- **Environment Management:** Dotenv

## 3. Architecture

---

The backend follows a Controller-Service-Model architecture: - `/src/routes`: Defines API endpoints and maps them to controllers. - `/src/controller`: Contains the business logic for each feature. - `/src/models`: Defines MongoDB schemas using Mongoose. - `/src/middleware`: Custom middleware for authentication and validation. - `/src/utils`: Helper functions (logging, emailers, analytics).

## 4. API Endpoints

---

- **Auth:** `/api/auth` (Login, Register, Profile)
- **Products:** `/api/products` (List, Detail, Admin management)
- **Cart:** `/api/cart` (Manage user shopping carts)
- **Orders:** `/api/orders` (Checkout, History, Management)

- **Analytics:** Internal logging for product and user activity.

## 5. Security and Middleware

---

- **Auth Guard:** Middleware that validates JWT tokens on protected routes.
- **CORS:** Configured for cross-origin requests from the frontend.
- **Password Hashing:** Uses `bcrypt` to securely store user credentials.

## 6. Core Logic

---

- **Checkout Process:** Validates inventory, calculates totals, creates order records, and clears the user's cart.
- **Analytics logging:** Asynchronous logging of user actions and product views to track engagement.