NutriStore sample code for POC - Technical Documentation (v1.0)

# **Project Overview**

NutriStore is a full-stack e-commerce platform tailored for nutritionists and dietitians to sell wellness products, meal plans, supplements, and more. It features admin dashboards, secure order handling, dynamic coupons, SEO-optimized pages, and reordering functionality.

# **System Architecture**

|  |  |
| --- | --- |
| **Layer** | **Technology** |
| Frontend | React (Vite) + Tailwind CSS |
| Backend | Node.js + Express |
| Database | MongoDB Atlas |
| Authentication | JWT Tokens |
| Payments | Stripe (optional setup) |
| Hosting | Vercel (Frontend) + Render / Railway / DigitalOcean (Backend) |

# **Folder Structure**

root/  
├── backend/  
│ ├── controllers/  
│ ├── models/  
│ ├── routes/  
│ ├── middleware/  
│ ├── config/  
│ ├── server.js  
├── frontend/  
│ ├── src/  
│ │ ├── components/  
│ │ ├── pages/  
│ │ ├── context/  
│ │ ├── App.jsx  
│ │ ├── main.jsx  
│ ├── public/  
├── .env  
├── package.json  
└── README.md

# **Tech Stack**

* **React** (Frontend UI)
* **Vite** (Frontend Bundler)
* **Tailwind CSS** (Styling)
* **React Router** (Routing)
* **React Helmet Async** (SEO)
* **Redux Toolkit** (State Management)
* **Axios** (API Requests)
* **Node.js + Express.js** (Backend API)
* **MongoDB + Mongoose** (Database)
* **JWT** (Authentication)
* **Stripe** (Payments Integration)
* **React Toastify** (Notifications)

# **Environment Variables**

## **Backend .env**

NODE\_ENV=development  
PORT=5000  
MONGO\_URI=your\_mongodb\_connection\_string  
JWT\_SECRET=your\_secret\_key  
STRIPE\_SECRET\_KEY=your\_stripe\_key

## **Frontend .env**

VITE\_API\_URL=http://localhost:5000

# **Setup Instructions**

## **1. Backend Setup**

cd backend  
npm install  
npm run dev

Backend will run on <http://localhost:5000>.

## **2. Frontend Setup**

cd frontend  
npm install  
npm run dev

Frontend will run on <http://localhost:5173>.

# **Deployment Instructions (Production)**

### **Backend:**

* Host using **Render**, **Railway**, or **DigitalOcean**.
* Set environment variables on the hosting dashboard.
* Connect your GitHub repository or upload zipped files.

### **Frontend:**

* Deploy frontend to **Vercel**.
* Set VITE\_API\_URL to your backend API endpoint.

# **API Endpoints**

|  |  |  |
| --- | --- | --- |
| **Endpoint** | **Description** | **Access** |
| GET /api/products | Get all products | Public |
| GET /api/products/:id | Get product by ID | Public |
| POST /api/products | Create new product | Admin |
| PUT /api/products/:id | Update product | Admin |
| DELETE /api/products/:id | Delete product | Admin |
| POST /api/users/login | Authenticate user | Public |
| POST /api/users/register | Register new user | Public |
| GET /api/users/profile | Get user profile | Private |
| POST /api/orders | Create new order | Private |
| GET /api/orders/myorders | List user’s orders | Private |
| POST /api/orders/:id/reorder | Reorder past order | Private |
| POST /api/coupons | Create a new coupon | Admin |
| GET /api/coupons | List all coupons | Admin |
| DELETE /api/coupons/:id | Delete a coupon | Admin |

# **Admin Features**

* Manage Products
* Manage Users
* View Orders
* Create & Manage Coupons
* Analytics Dashboard (upcoming)

# **User Features**

* Browse Products
* Add to Cart
* Checkout
* Order History
* Buy Again
* Wishlist (upcoming)

# **Suggested Improvements**

* Add multi-currency support
* Add delivery status tracking
* Integrate Google Analytics
* Add customer reviews & ratings
* Enhance SEO with structured data

# **Scaling Tips**

* Use Redis for session caching
* Optimize MongoDB indexes
* Implement rate limiting
* Lazy-load images on frontend
* Server-side rendering (SSR) for critical pages