**Sweet Shop Backend API**

This project is a backend API for a sweet shop, designed to manage user authentication, sweet inventory, and customer orders. The API is built with Node.js and Express, following a Test-Driven Development (TDD) approach to ensure a robust and well-tested codebase.

**Features**

* User authentication (Registration, Login)
* CRUD operations for sweet inventory (Admin only)
* Real-time stock management
* Order placement and tracking
* General Notifications
* Search bar for both user and admin

**Setup and Running the Project**

Follow these instructions to set up and run the Sweet Shop project on your local machine. This project consists of a backend (Node.js/Express) and a frontend (React).

**1. Backend Setup**

1. **Navigate to the backend directory:**
2. cd sweet-shop-backend
3. **Install dependencies:**
4. npm install
5. **Create a .env file** Create a file named .env in the sweet-shop-backend directory and add your environment variables.
6. MONGO\_URI=your\_mongodb\_connection\_string
7. JWT\_SECRET=a\_strong\_secret\_key

*Replace your\_mongodb\_connection\_string with your actual MongoDB connection string.*

1. **Run the backend server:**
2. npm start

The backend server will run on http://localhost:5000 (or the port specified in your code).

**2. Frontend Setup**

1. **Navigate to the frontend directory:**
2. cd sweet-shop-frontend
3. **Install dependencies:**
4. npm install
5. **Run the frontend application:**
6. npm start

The frontend application will open in your browser, typically at http://localhost:3000.

**My AI Usage**

**AI Tools Used**

This project utilized a Gemini, chatgpt to assist with various aspects of the development process.

**How I Used the AI**

I used Gemini as a collaborative partner and a debugging assistant, particularly for my TDD workflow. Specifically, I used the AI to:\

This project was built with my own coding and guidance from AI tools (ChatGPT,Gemini).

**- AI helped me generate the initial Express + MongoDB backend boilerplate.**

- AI assisted in scaffolding the React frontend with Bootstrap and React-Toastify.

- I customized the code, added validations, refactored structure, and wrote my own improvements.

- All code was reviewed, tested, and debugged by me.

- AI suggestions were used as a coding assistant, not as a replacement. Nothing was copy-pasted from external repositories.

**My Reflection on AI's Impact**

Using an AI assistant had a significant impact on my workflow. It acted as an effective pair programming partner, helping me stay on track with the TDD methodology. The AI's ability to quickly identify and explain errors in my Git workflow was particularly valuable, saving me time and frustration. By using the AI to generate initial boilerplate and tests, I was able to focus more on the core business logic and architectural design of the application. It helped me maintain a disciplined and clean development process from start to finish.