**Canteen Ordering System**

**Project Overview**

This project illustrates a Canteen Ordering System which is a web-based platform designed for students and staff to browse the canteen menu, customize orders, place them in a cart, and track the status of their orders in real-time. The system features user authentication, a menu display with customization options, and an interactive ordering system. Users can also enjoy a loyalty program where they earn points for each order, redeemable for discounts. This project provides an intuitive, seamless ordering experience, making it easier for users to manage their orders online.

**Deployment Link**

* [Canteen Ordering System](https://canteen-ordering-system-two.vercel.app/login)

**Login Details**

For authentication, you can use the following test credentials:

* **Username**: vanessa.umenze@acity.eud.gh
* **Password**: webtech2024

Note: *However, it is also possible for the user to register an and use their own credentials to log in as all these are stored in the database*

**Feature Checklist**

* **User Registration & Profile Management**  
  Secure registration and login system with user profile management.
* **Menu Display & Customization**  
  Display menu items with options for customization (e.g., extra toppings).
* **Ordering System with Cart**  
  Add items to the cart, view order totals, and place orders.
* **Order Tracking & Loyalty Program**  
  Track the order status in real-time and earn loyalty points with each purchase.

**Technologies Used**

* **Frontend**: React.js, Tailwind CSS, Framer Motion (for animations)
* **Backend**: Express.js, PostgreSQL
* **Authentication**: JWT (JSON Web Tokens) for secure login and session management
* **State Management**: React’s useState and useEffect hooks for managing application state

**Project Structure**

**Frontend**

* **Login**: Allows users to log in using their credentials.
* **Menu**: Displays a list of available menu items with pricing and customization options.
* **Cart**: Displays the items currently in the user's cart, allowing them to remove items.
* **Order Tracking**: Tracks the order status based on the order ID.
* **RecommendationCard**: Displays individual food items and allows users to add them to their cart with customizations.

**Backend**

* **Authentication**: JWT-based authentication with secure login routes.
* **Cart Routes**: Add, fetch, and delete cart items associated with a user.
* **Database**: PostgreSQL database to manage user data, cart items, and menu data.

**Installation Instructions**

To run this project locally, follow these steps:

**1. Clone the repository**

git clone https://github.com/vanessa-196/canteen-ordering-system.git

cd canteen-ordering-system

**2. Install dependencies**

*For the frontend:*

cd client

npm install

*For the backend:*

cd server

npm install

**3. Set up environment variables**

Create a .env file in both the frontend and backend directories and add the necessary variables such as API keys, database connection strings, and JWT secret for authentication.

**4. Start the application**

*For the frontend:*

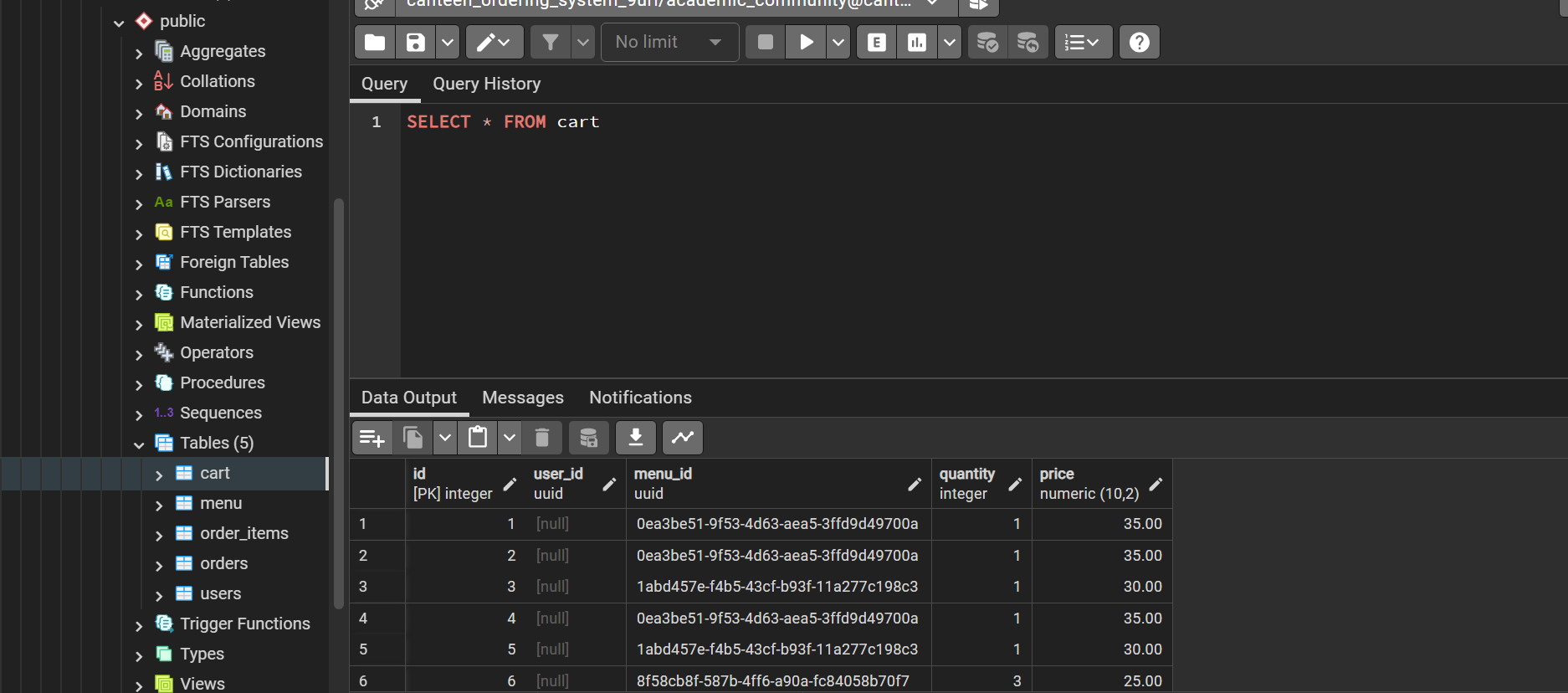
npm start

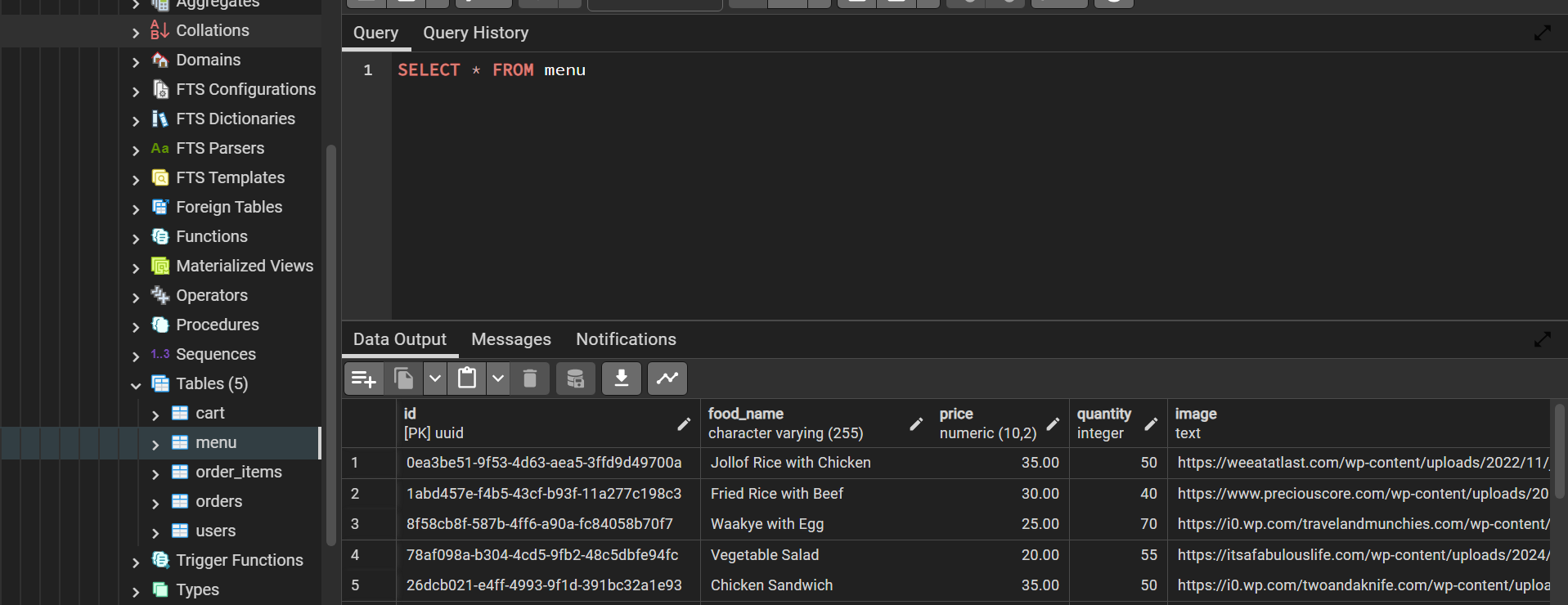
*For the backend:*

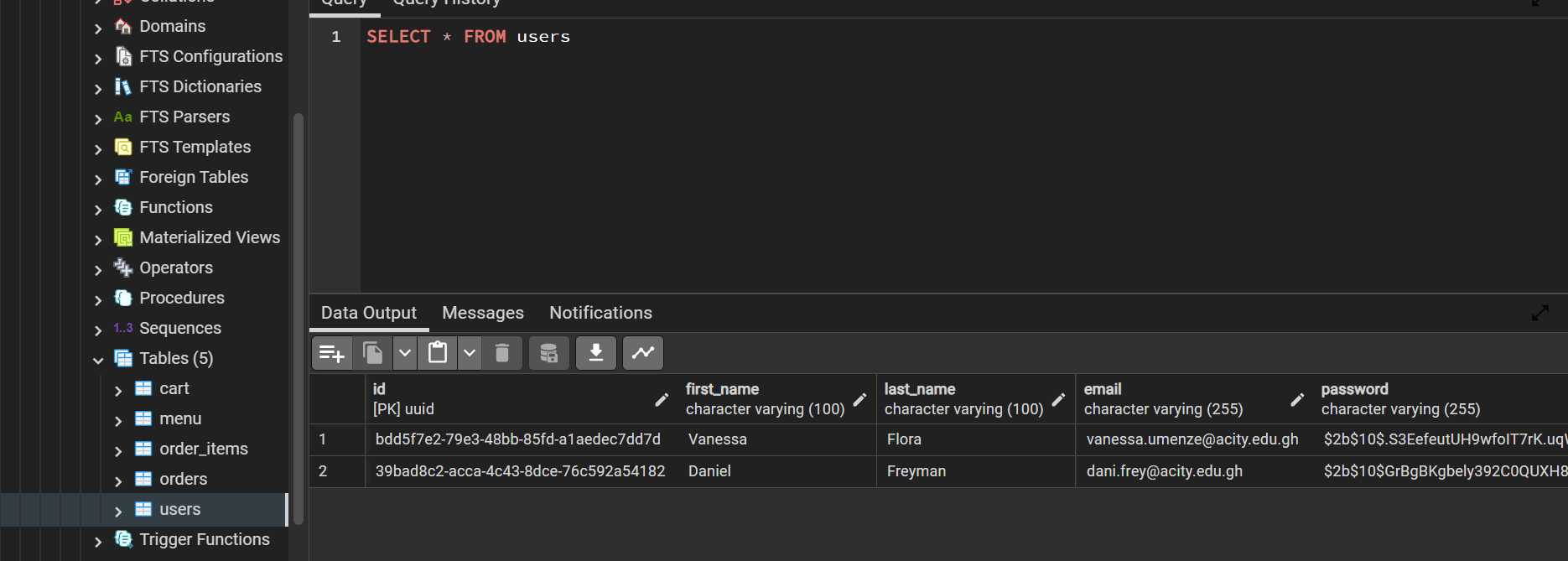
npm start

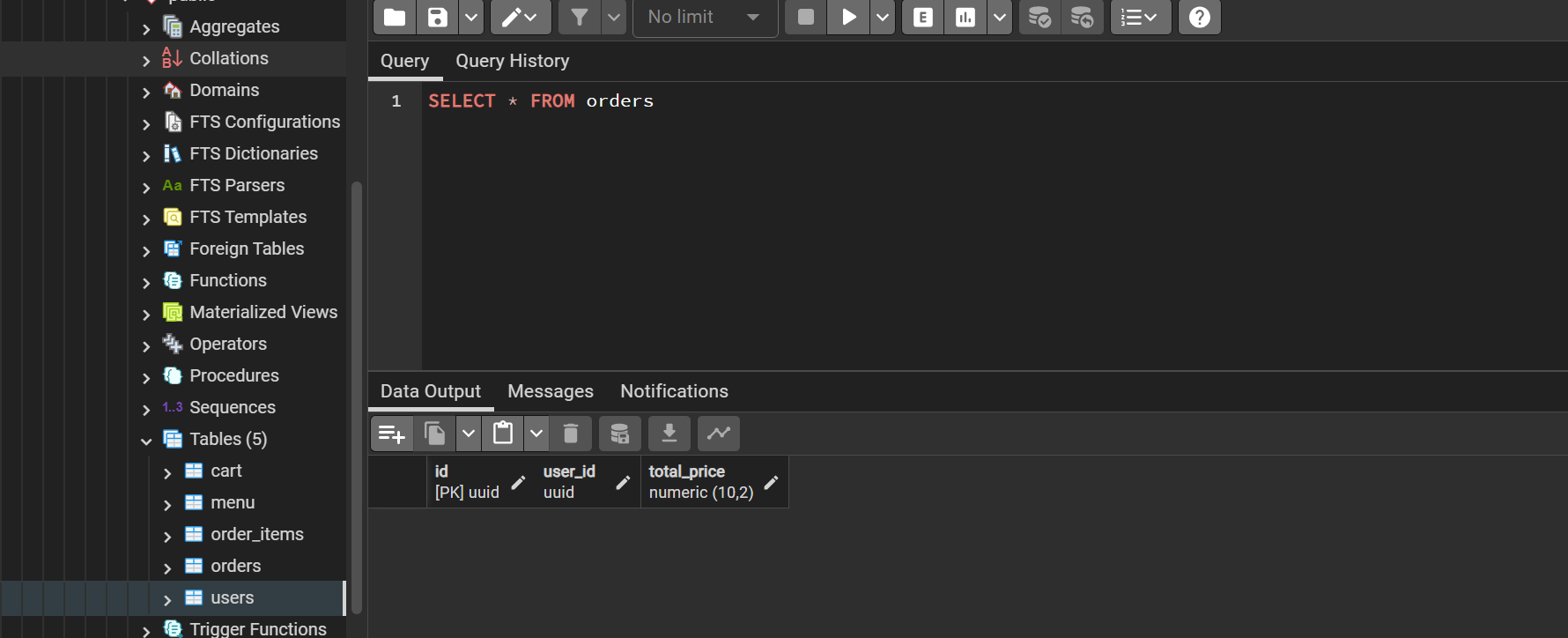
Your application should now be running on your local host

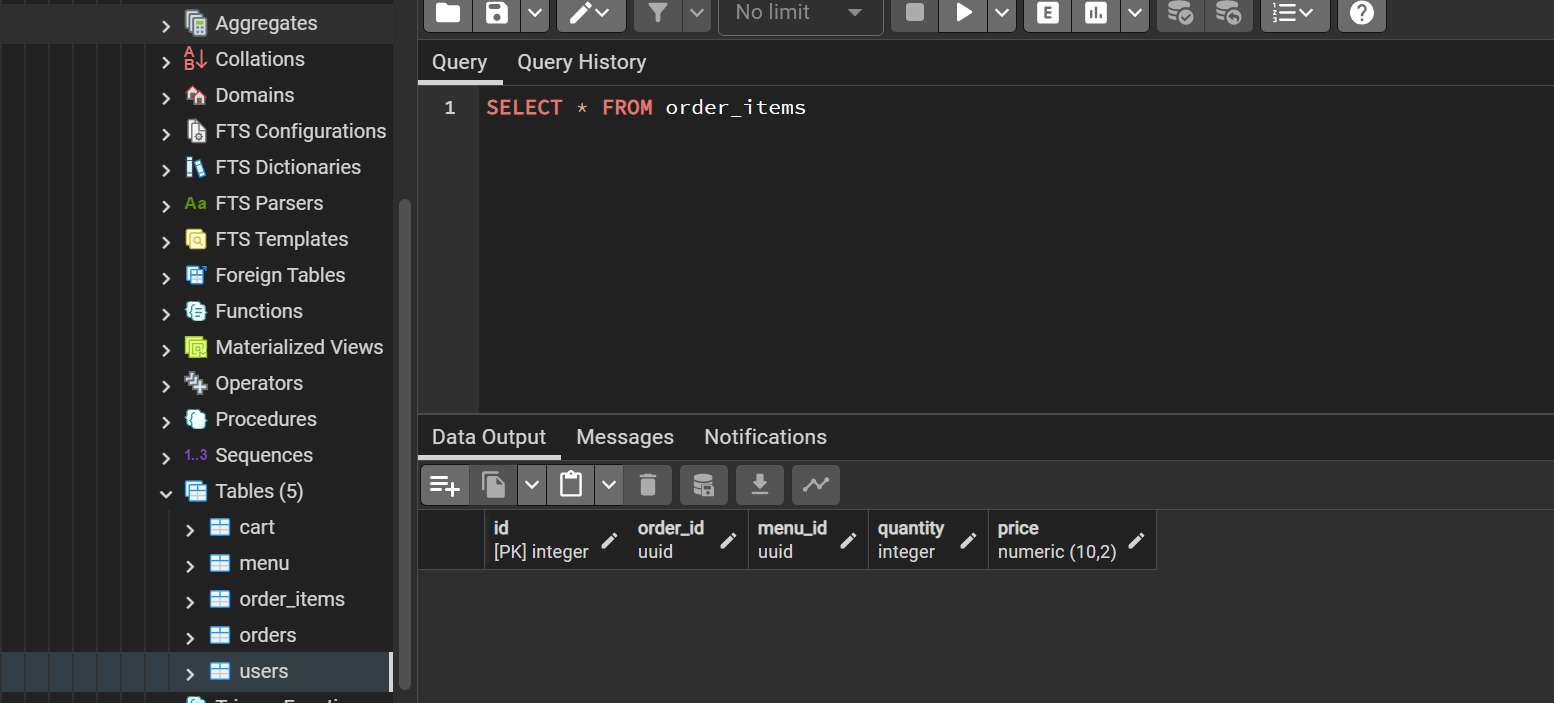
**Screenshot of Database:**













**Additional Features for Improvement**

**Conclusion**

This project demonstrates my skills in full-stack development, from designing and implementing user interfaces with React to building secure backend APIs with Express.js and PostgreSQL. While most of the core features are functional, further work is required to ensure full cart functionality and fix any issues related to item addition and CORS handling.

I look forward to continuing work on this project to refine its features and resolve any outstanding issues.

**Future Enhancements**

* Order History: Users will be able to view their past orders and reorder easily.
* Admin Panel: An admin interface for managing menu items, prices, and user orders.
* Payment Integration: Adding payment gateways for processing transactions within the app.