

CS101: Web Stack Fundamentals

Lab Assignment 4: “The Pantry Tracker” (Drizzle Edition)

Goal

Build a Single Page Application (SPA) to track food inventory using the PERN stack (PostgreSQL, Express, React, Node.js) and Drizzle ORM.

1 Database & Schema Configuration

- **System:** PostgreSQL managed via Drizzle ORM.
- **Table Requirement:** You need a single table (`pantry_items`) capable of storing:
 - Unique ID
 - Item Name (e.g., “Milk”)
 - Quantity (e.g., 2)
 - Unit (e.g., “liters”)
 - Expiry Date
 - Date Added
- **Migration:** The database schema must be generated and synced using Drizzle’s migration tools, not created manually in SQL.

2 Backend API (Express)

You need to expose a REST API with the following capabilities:

- **List Inventory:** An endpoint to retrieve all items. The data sent to the frontend must be sorted so items expiring soonest appear first.
- **Add Item:** An endpoint to accept new item details. Upon success, it must return the newly created item data (including the generated ID) to the client.
- **Update Item:** An endpoint to modify a specific item (specifically for changing quantity).
- **Remove Item:** An endpoint to permanently delete an item by its ID.

3 Frontend Interface (React)

- **Dashboard View:** On page load, fetch and display the inventory.

- **Visual Alert:** Any item that has passed its expiry date must be visually distinct (e.g., red background or warning icon).
- **Input Mechanism:** A form to add new items. When submitted, the list should update immediately without refreshing the page.
- **Interaction:**
 - Users must be able to delete an item directly from the list.
 - Users must be able to increment/decrement the quantity of an item directly from the list.

4 Deliverables

- A functioning backend that connects to PostgreSQL using Drizzle.
- A functioning frontend that consumes your API.
- **Persistence:** Data must survive a server restart.

Suggested First Step: Initialize your project folder, install your dependencies (`express`, `drizzle-orm`, `pg`, `react`), and set up your `drizzle.config.ts`.