**Grocery Shop Till Documentation**

**Overview**

The Grocery Shop Till project is a Java-based application made to simulate the operations of a grocery store. It includes functionalities such as adding, deleting and retrieving items and special deals, and scanning items for purchase.

**Installation**

1. **Clone the repository**

git clone https://github.com/GabrielMindev/grocery-shop-till.git

cd grocery-shop-till

1. **Build and Run:**

./mvnw clean install

./mvnw spring-boot:run

**Usage - Endpoints**

**Get All Items:**

**URL:** http://localhost:8080/grocery/items

**Method:** GET

**Description:** Retrieve all grocery items in the store.

**Get All Deals:**

**URL**: http://localhost:8080/grocery/deals

**Method:** GET

**Description:** Retrieve all special deals available.

**Add Item:**

**URL:** http://localhost:8080/grocery/items

**Method:** POST

**Description:** Add a new grocery item to the store.

**Delete Item:**

**URL:** http://localhost:8080/grocery/items/{name}

**Method:** DELETE

**Description:** Delete grocery item from the store.

**Add Deal:**

**URL:** http://localhost:8080/grocery/deals

**Method:** POST

**Description:** Add a new special deal.

**Delete Deal:**

**URL:** http://localhost:8080/grocery/deals/{type}

**Method:** DELETE

**Description:** Delete special deal.

**Scan Items:**

**URL**: http://localhost:8080/grocery/scan

**Method:** POST

**Description:** Scan a list of items and calculate the total cost.

**UI**

The project includes a basic user interface for interacting with the above endpoints. It consists of index.html page, along with script.css forming the style of the page and script.js file which serve as the client-side responsible for handling user interactions, making requests to the server, handling the response and DOM manipulating.

**Folder Structure**

The structure consists of several packages divided by responsibilities.

-**controller/ :** Contains the controller classes responsible for handling HTTP requests and managing the flow of data between the user interface and the service layer.

-**model/:** Contains entity classes representing the data structure of the application, like GroceryItem and SpecialDeal.

**-service/:** Contains service class that implement the business logic of the application.

**-src/main/resources/:** This directory contains all non-Java resources like the index.html, style.css and script.js files

**-pom.xml-**  Maven Project Object Model file that defines the configuration and dependencies for the project.

**Code Documentation**

Code itself includes comments for the more complex logic.

**Dependencies**

-Java 11

-Spring Boot

-Maven

-Lombok – added for reducing the boilerplate code