Author: Nishkarsh Sharma
Roll Number. : 21f1000053
Email id: 21f1000053@ds.study.iitm.ac.in

Description:

The Grocery Store offers a variety of products for purchase online. The admin creates and manages the categories and products. Customers can quickly check and purchase products, search for products, and add them to their shopping carts, which the customer can update at any time.

Technologies Used:

1. Flask – For API
2. JWT Security – involves the use of tokens for authentication and authorisation.
3. CORS – Cross-Origin Resource Sharing
4. Node Modules-for generating Frontend and CSS integrated with them
5. SQLite and SQLAlchemy - For data storage.
6. Vuejs and Vue-toast-notification - For UI and For Pop-up Messages
7. Redis – For Caching
8. Celery integrated Redis for Sending Activity reports, Reminder(Batch-jobs) and export CSV for store managers.

DB Schema Design:

The following are the DB tables with columns and constraints. Each table has a unique identifier (id as primary key).

1. Users:
    a. id (Primary Key, Integer)
    b. name (String, max length 1000)
    c. username (String, max length 100, Unique)
    d. password (String, max length 100)
    e. role (Numeric)
2. Categories:
    a. id (Primary Key, Integer)
    b. name (String)
3. Product:
    a. id (Primary Key, Integer)
    b. name (String, max length 100)
    c. category_id (Foreign Key, References Categories. id)
    d. expiry_date (Date)
    e. manufacture_date (Date)
    f. price (Float)
    g. unit (String)
    h. quantity (Float)

    i. created_at (DateTime)
4. Order:
    a. id (Primary Key, Integer)
    b. user_id (Foreign Key, References User. id)
    c. product_id (Foreign Key, References Product. id)
    d. quantity (Float)
    e. total (Float)
    f. status (String)
    g. product_name (String, max length 100)
    h. product_price (Float)

API Design: APIS have been for the following:

1. CRUD operations for Categories
2. CRUD operations for Products
3. Validation of all input fields - text, numbers, dates etc.
4. Backend validation before storing.

Architecture and Features:

Architecture:

The Python code for the project is organised into 5 files ➖

1. App.py - Contains the code for starting the grocery store application.
2. celery_config.py - Contains the setup of the Redis celery server.
3. Reminders.py - Implementation of daily reminders
4. tasks.py -Implementation of CSV generation and monthly report task(celery).

Features implemented:

1. User Signup and Login (Token Based).
2. Separate Signup and login for admin and store manager.Only one admin with admin key.
3. First of all the Store Manager has to sign up and the admin can approve the Store-manager. After that, the store manager can log in with the store manager key.
4. Creation, update and deletion of Categories by admin.
5. Creation, update and deletion of Products by Store-manager.
6. The store manager can export a CSV File regarding his inventory.
7. Users can add multiple products at one time.
8. Users can search any product and category by name.
9. If the User is not buy anything from the store We will send them reminder emails.
10. Monthly report for users for their purchased items and expenditure.
11. Users can update any product quantity directly and their amount is also incremented in the cart.
12. Users can directly delete any item from the cart.
13. If any product is finished so you can not add it to the cart. Show “Out of stock”.
14. After Placing the order the placed order will show on the profile section.

Video Link:

Project_Video_Mad2