

Author:

Animesh Sharma

Roll Number. : 21f1005932

Email id: 21f1005932@ds.study.iitm.ac.in

Description:

The Grocery Store is an online platform for buying various food products in different categories. The categories are created and managed by the store admin. Products are managed by the store manager. He can Add, Update and Delete multiple products in categories. Buyers can easily search their categories and respective products.

Technologies Used:

1. Flask – For API
2. JWT Security – involves the use of tokens for authentication and authorization.
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 - For UI
7. Redis – For Caching
8. Celery integrated redis for Sending Activity reports, Reminder(Batch-jobs) and export CSV for store managers.

DB Schema Design:

1. The following are the DB tables with columns and constraints. Each table has a unique identifier (id as primary key).
 - **Products:** (Columns)
 1. Product_id: Product id(Non-Nullable)
 2. Product_name: Name of product(Non-Nullable)
 3. Product_price: Price of product(Non-Nullable)
 4. Product_quantity: Quantity of product(Non-Nullable)
 5. Product_manufacturing date: Manufacturing date of product(Non-Nullable)
 6. Product_expiry date: Expiry date of Product(Non-Nullable)
 7. Product_category_id: Managed by category_id(Foreign key, Non-Nullable)
 8. Product_created_by :By store manager(Foreign Key)
 - **Cart:** (Columns)
 1. Cart_items: cart can have multiple products
 2. Cart_count: Count of items in the cart
 3. Cart_last purchased: Save date when user buys something.
 4. cart_expenditure: Total cart expense

- **Cartitems:** (Columns)
 1. Cartitems_cart_id: Cart to which cart item belongs(Foreign key, Non-Nullable)
 2. Cartitems_product_id: Product associated with cart item(Foreign key, Non-Nullable)

API Design: APIS have been for the following:

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

Architecture and Features:

The python code for the project is organized into 2 files –

- App.py: It contains all my code and the logic for all API Implementation.
- Node_modules:Contains all the frontend code and styling.

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 login 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. 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 categories by name.
9. If User is not buy anything from the store so We will send them reminder mails.
10. Monthly report for users for their purchased items and expenditure.
11. Users can update any product directly and their amount is also incremented in cart.
12. Users can directly delete any item from the cart.
13. If any product is finished so you can not add in cart.Show “Out of stock”
14. Final checkout page and total amount is shown before placing order.

Video Link:

[mad-2.mp4](#)