Author :

HIMANSHU KUMAR

21f3001746

21f3001746@ds.study.iitm.ac.in

I am a final year ECE engineering student with a strong interest in data science and machine learning, working towards becoming a data scientist.

Description :

This project is a platform for making online purchases for Grocery items like Blinkit, BigBasket, Zepto etc. Here user can add item to his/her personal cart and make purchases for different items collectively like any usual grocery application. There are two more levels one is Manager and Admin with divided actions along with Batch jobs.

Technologies used :

1. Flask - for API
2. VueJS CLI - for UI.
3. Jinja2 templates - sending mails
4. Bootstrap
5. SQLite and SQLAlchemy for data storage.
6. Redis and Celery for Batch jobs

DB schema design :

a. The database has several models/tables created: Users, Userdata, Shop, Category and Requests. Each table has different attributes and helper functions.

b. The database is designed to store user information, and the information about the items in the Shop and the category they belong to.

c. Structure and details of the columns :

1. 1. Users: This table stores information about the users and managers of the application, including their email Id , passsword , userid, role.
2. 2. Userdata: Contains information about the user when did he logged in last, current cart items, purchase history.
3. 3. Shop: This table stores information about the items in the store like itemid , their category name , stock left, price , image, stock sold , etc
4. 4. Category: This table store all the category name and category id.
5. 5.Requests: This table store all the requests from the manager to teh admin it can be :-

5.1) to delete a category
5.2) to edit a category
5.3) to add a category
5.4) to accept a joining request of a new manager

API design :

The API design for this project follows a RESTful architecture, which means that it uses HTTP requests to access and manipulate resources identified by URLs. The endpoints are grouped into several categories, each corresponding to a specific feature of the application.

Architecture and Features :

a. Features implemented include :

1. User and Manager registration and login (using username and password) - implemented using JWT (token-based authentication)

2. What user can do:
   - ☑ Ability to buy multiple products from one or multiple sections.
   - ☑ Ability to search multiple products or categories based on their name or price.
   - ☑ Ability to see out of stock products that are not available.
   - ☑ Ability to see the total amount to be paid for the transaction.

3. What Store Manager can do:
   - ☑ Ability to add or edit multiple products from one or multiple Categories.
   - ☑ Ability to remove a product with a conformation.
   - ☑ Request admin to add new, edit or delete multiple categories.

4. Well-designed PDF reports
   - ☑ HTML or PDF reports

5. Handling images:
   - ☑ Automatically getting new images when creating a new product.

6. What Admin can do:
   - ☑ Ability to add new, edit or delete multiple categories.
   - ☑ Ability to accept or remove any Store manager request relate to category management.

7. Daily Reminder Jobs
   - ☑ Reminders using email.

8.Scheduled Job - Monthly Engagement Report
   - ☑ Send monthly progress report using mail

8. User Triggered Async Job
   - ☑ Export blog details as CSV

- ☑ Send an alert when done.

9. Performance and Caching
   - ☑ Add cache to increase performance
   - ☑ Add cache expiry

Additional features :

1. **Search** functionality:
   - ☑ can filter items by price.
   - ☑ can search items by name.

2. Handling images :
   - ☑ Add images to a item with the backend server live time.

3. Well designed PDF reports
   - ☐ HTML or PDF reports

4. Styling and Aesthetics - done

Video :

Drive link →

https://drive.google.com/file/d/17o3_Up9dK3SnDA1Qhs5U9FD1dYdB5jd1/view?usp=sharing

PlayVideo