Author:
Name – Rituparno Sen
Roll - 21f1004275
Email - 21f1004275@ds.study.iitm.ac.in

Description :

This project is a multi-user (one/multiple admin/store manager and other users) app for buying grocery from a grocery store where admin can create and manage(edit and/or delete) categories and products and other users (customers) can create account and log in to buy products from the shop. Any guest can check or search for products without the need to login. Logging in is necessary for adding to cart or checkout orders.

Technologies used :

1. Flask - for application code
2. Jinja2 templates and Bootstrap for HTML generation and styling.
3. SQLite and SQLAlchemy for data storage.
4. Flask-Login, Flask-WTF, WTForms, bcrypt – for convenient and authenticated login system

   And all other necessary dependencies for the aforementioned technologies.

DB schema design :

a. The database has several models/tables created: User, Section, Product, Cart, , Cart-item, Shopping-list. Each table has different attributes and helper functions.

b. The database is designed to store user username & password, category & product details and product back refers to category, cart, cart-item, shopping-list details and relationships between them for the smooth functioning of the application.

c. Structure and details of the columns :

1. User : This table stores information about the users , including their username, password, assigned roles, shopping carts and shopping list .
2. Section : Section table stores the name and other details of Section and also store relationship with associated product.
3. Product : This table stores detailed information about product including their name, pricing per unit, type of unit, quantity, MFD ,Exp date,and category. Also store information of relationship with cart item, shopping list.
4. Cart : This table represent shopping carts associate with users and contain cart item with a relationship .
5. CartItem : This table stores individual items with shopping carts , including product details and quantities.
6. ShoppingList : This table stores the information about shopping lists, including the user who created it , checkout date ,product in the list and quantities .

API design : Due to my insufficient knowledge in api. I could not handle them well so I did not use api for this project.

Architecture and Features :

a. The project is organized using the Model-View-Controller (MVC) architecture, with the controllers handling logic and routing, templates for displaying views, and models for interacting with the database. The app can be run from main directory using the 'run.py' file. The folder 'app' contains the python scripts and templates. Folder 'instance' is where the main database is located.

b. Features implemented include :

1. Admin(store manager) can login (using email and password which is set by developer. Developer also can add more admins).
2. Admin dashboard
    - Manage category
        - ✓ Add new category
        - ✓ Edit category
        - ✓ Delete category
        - ✓ View products under category
    - Manage products
        - ✓ Add new product
        - ✓ Edit product details
        - ✓ Delete product
        - ✓ Add quantity to product
        - ✓ Add rate per unit to product
        - ✓ Add unit type to product
    - Logout admin.
3. User (customer) can create account using username,email and password and then use email and password to login. Emails are supposed to be unique and thus primary key for the user
4. User dashboard
    - ✓ Search for category, product with different options like manufacture date, expiry date, price.
    - ✓ Basic view of available products and categories.
    - ✓ Add to cart & Buy(Only available to logged in user)
    - ✓ View cart
    - ✓ Checkout
    - ✓ View shopping history
    - ✓ Logout
5. Validation
    - ✓ Server side validation with python, WT forms
    - ✓ Client side validation with HTML

    These features are implemented using functions created inside particular route for each functionality.

Link to video:

https://drive.google.com/file/d/1-eYMzr_hsil9qizjWR3vHabs18eLw3Qf/view?usp=drive_link