Grocery Store V1 Application

Author

V CHARAN TEJA REDDY
21F1006486
21f1006486@ds.study.iitm.ac.in
I'm a recently graduated student in B.Sc. Maths, Electronics and Computer Science. I'm interested in Data Science and I'm sure I'll learn a lot in IITM BS in DS & Applications degree.

Description

In this project, we are supposed to build a Grocery Store Application which is used to purchase groceries. The project should have at least one admin and users who can buy the displayed groceries. We need to have login for user/admin, sign-up for users, category management & product management by admin and search & buy features for the users.

Technologies used

Flask : Application routing, page rendering & redirecting, flash messages, managing sessions and creating blueprints
SQLite : Database
Flask-SQLAlchemy : For database connection with Flask Application & to manage it
Flask-login : To control access to pages and functions
Flask_bcrypt : To hash passwords before storing them in database
Pillow : To save images
Jinja2 : For html template generation
Bootstrap : For a clean UI

DB Schema Design

User : id, username, password_hash & cart (used for one-to-many relationship with Cart)
Cart : id, user_id(User foreign key), prod_id(Product foreign key), product_quantity
Product : id, product_picture, total_quantity, rate_per_unit, manufacture_date, unit_id(Unit foreign key), category_id(Category foreign key) & cart (One-to-many relationship with Cart)
Category : id, category_name, category_picture & products(One-to-many relationship with Product)
Admin : id, admin_name, & password_hash (No relationships as Admin just creates and edits but doesn't need to be associated with any of them)
Unit : id, unit_name & products(One-to-many relationship with Product)

API Design

I haven't created APIs

Architecture and Features

In the root directory of the application, there is requirements.txt, app.py which runs the application, instance folder which contains database, and application folder. The application folder contains __init__.py, auth_admin.py, auth_user.py, models.py and views.py files. Along with these, it also contains static folder inside which there is css folder with style.css and images folder with images and templates folder which contains all the html files(admin templates are in admin folder inside templates folder).

I have implemented all the core requirements along with validation before storing in the database. The core requirements are :

- Admin login and User login/sign-up
- Category Management by Admin
- Product Management by Admin
- Search & Cart multiple products
- Buy products from multiple categories

I first started the project by designing the models for the database. Then I created the application structure to get an idea of the role of each python file in the backend. After that I created the base html template which the application is going to follow and then extended it to other html files for every webpage. In the backend, I started with the admin functionalities following CRUD for categories, products and units. I made sure only the admin can access those routes. Then I went on with user functionalities like search, cart & buy. I made sure people can view the categories and products without logging in but need to login to cart/buy products.

Video

https://drive.google.com/file/d/1Q40dHwBtHEI474PGyxoHfNT6_No-4Kpm/view?usp=sharing