Author

Aditya Prashant Dayal
21f1005489
21f1005489@ds.study.iitm.ac.in

I am Aditya Dayal, currently studying CS in my college. I am perusing the Diploma Level of this Data Science Degree program. I like to use my knowledge and learnings to build something which is helpful to society.

Description

This project consists an application which can be used in Grocery Stores. Users can register, login, buy products, search products, cart them and pay accordingly. Admin can manage the inventory and is able to do CRUD operations on products as well as on their categories.

Technologies used

1. Flask for application code
2. Flask-SQLAlchemy for connecting application to database
3. Jinja2 templates + Bootstrap for HTML generation and styling
4. SQLite for data storage

DB Schema Design

Database has many Models and tables in it. Database consist three Models named User, Category and Product. Following is the structure of the database:

1. User: It stores the information of the user who is using the application. User has attributes like id, username and password. Id is the primary key and constraints used are unique and not nullable
2. Category: Category stores the categories of products in the application. It has two attributes as id, category_name and items as a relationship to backreference Product table as section. Constraints used are unique and not nullable.
3. Product: This table is used to store info of products. It has attributes named id, product_name, unit, m_and_e_date, price, quantity and cat which have foreign key with category id. Constraints are unique and not nullable.

I used this schema because it's simple to implement and form perfect relationships of categories and products. Backreference minimises the redundancy in the database.

API Design

I haven't used API functionalities in this project. (it's in recommended/graded not in core)

Architecture and Features

Project Organization:

In this project, I used MVC (Model-View-Controller) architecture. Database is the Model which stores the data and its integrity. All the HTML templates acts as the view part where user can interact with the application. Routing and logical connecting and rendering the pages comes under the controller pages. My project consists app.py file having imports, routes, models, etc. templates folder have all the HTML files and static folder have CSS file and instance folder have database instance.

Features Implemented:

1. Admin login and user registration and login page using HTML forms.
2. Admin's dashboard: (CRUD Operations)
    a. Admin can Create and Read the categories and products
    b. Admin can Update the category or product information
    c. Deletion the category or product can be done with Admin's confirmation
    d. Admin can create products in any categories
3. User Dashboard:
    a. User can browse through various categories and products
    b. User can buy product and can add that product to cart
    c. It shows out of stock if inventory of that product is zero
4. User can search product by their name or category
5. User can search product by its price and manufacturing date, etc
6. While buying product, user can provide integer or decimal values for quantity
7. Cart:
    a. User can add multiple products to the cart from any categories
    b. User can define required quantity of products in the cart
    c. User can remove item from cart
    d. User can review and buy single item from cart

Additional features:

1. User profile:
    a. User can see their id and username
    b. User can change their username and password
2. Summary page before final buying of product is done

Video Link

https://drive.google.com/file/d/1GzN0PR50C2Q53I3wCMr7yimqo4-AyL-s/view?usp=drive_link