

Author

Dodda Abhiram

22f2000660

22f2000660@ds.study.iitm.ac.in

I like mathematics and using it in designing computer algorithms. Likes to build applications that involves data.

Description

The objective of the project is to build a grocery store app with one app and multiple users using Flask framework and concepts of Modern Application Development – 1 course. Project deals with sqlite database to store grocery store data and user data. The name of the project **SmallBasket** inspired from BigBasket.

Technologies used

The main technologies used in this application are flask-sqlalchemy, flask-restful, SQLite and Python. **Flask-sqlalchemy** is used to interact with **SQLite** database that makes this application portable. **Flask-restful** is used to make REST APIs to handle inputs. **Jinja2** is used to create user interaction templates.

DB Schema Design

The app has SQLite named **database.sqlite3** database with 7 tables in the database folder. These tables store user and manager credentials, categories and products managers add, a table to keep track of user purchases, sales for managers and cart to keep track of user items.

| Table name | Column details |
|-------------------|---|
| managers | manager_id(integer, primary key), username(string), password(string) |
| users | user_id(integer, primary key), username(string), password(string), email(string) |
| categories | category_id(string, primary key), name(string), search(string) |
| products | product_id(string, primary key), category_id(string), name(string), search(string), price(numeric), stock(numeric), unit(string), fractal_allowed(string) |
| cart | cart_id(string, primary key), user_id(string), product_id(string), quantity(numeric), unit(string), price(numeric) |

| | |
|------------------|--|
| sales | product_id(string, primary key), category_id(string), quantity(numeric), sale(numeric) |
| purchases | Transaction_id(string, primary key), user_id(string), price(numeric), date(string) |

API Design

This application uses a total of 9 REST APIs to collect and process user data. All of the APIs use only **post** method as HTML forms does not support **put** and **delete** requests. The main tasks they take care of includes adding, editing and deleting of products and categories by managers, editing and deleting purchases of users and search feature.

Architecture and Features

The project starts with **start.sh** that is located in code folder (**GroceryStore** folder). README file contains information on how to start the application. Database interactions are taken care by models, located in **models.py** in models folder and helper functions, located in **helpers.py** in Helpers folder. Database and **db.py** file which creates instance of SQLAlchemy reside in database folder. All the APIs reside in **api.py** of API folder. HTML and CSS files are in **templates** and **static** folders respectively. Controllers are in **app.py**.

In the application only one manager exists. The manger can create categories which in empty by default, manager can add products under each category along with necessary information like stock, price etc. Manager is allowed and delete products and categories. If a non-empty category is deleted all the products that belong to the category are also deleted. Users can see all the categories and products added by managers in their respective dashboard. They can buy one quantity of any product by clicking Add to cart or add the quantity they like by selecting Buy Now. All the products they buy among any category are displayed in cart as they are saved in cart table. In cart they can edit and delete any purchase.

Video

<https://drive.google.com/file/d/1XBPMESzW6vGmP0E1-LYU2uL9czlh23Vx/view?usp=sharing>