Author

SAKTHI BALAN BALASUBRAMANIAN
21f3002711
21f3002711@ds.study.iitm.ac.in
A passionate 3rd year B.Tech ECE student also pursuing an IITM online BS degree in Data Science and Programming. Eager to explore the uncharted and expand my knowledge horizons.

Description

It is required to create an application where the users can select grocery items from different categories and buy them while the admin can add or delete the available and unavailable goods. Thus, its used in creating a platform between the seller and the customer of a grocery store through an application.

Technologies used

1. **Flask**: This provides the core framework for building the web application using the MVC (Model-View-Controller) architecture.
2. **Jinja2**: It is used for rendering dynamic HTML templates with placeholders that allow inserting data from the backend into the frontend.
3. **Flask-Login**: It is used to manage user authentication, session management, and restricted access to certain routes based on user roles (admin/user).
4. **Flask-SQLAlchemy**: This is used to simplify the interaction with the SQLite database by providing an ORM for defining database models and querying data using Python classes.
5. **Werkzeug**: This provides secure password hashing for storing passwords in the database and checking their validity during login.
6. **SQLite**: It is chosen as the database engine for its simplicity and suitability for small-scale applications.

DB Schema Design

My database consists of 5 different tables whose structures are defined as follows:

Admin table:

```
CREATE TABLE admin (
        admin_id INTEGER NOT NULL,
        email VARCHAR NOT NULL,
        first_name VARCHAR,
        password VARCHAR NOT NULL,
        PRIMARY KEY (admin_id),
        UNIQUE (email)
);
```

Cart table:

```
CREATE TABLE cart (
        cart_id INTEGER NOT NULL,
        quantity INTEGER NOT NULL,
        product_name VARCHAR,
        category_id INTEGER,
        user_id INTEGER,
        PRIMARY KEY (cart_id),
        FOREIGN KEY(product_name) REFERENCES product (product_name),
        FOREIGN KEY(category_id) REFERENCES categories (category_id),
        FOREIGN KEY(user_id) REFERENCES user (user_id)
```

```
);
Categories table:
CREATE TABLE categories (
        category_id INTEGER NOT NULL,
        type VARCHAR NOT NULL,
        PRIMARY KEY (category_id)
);
Product table:
CREATE TABLE product (
        product_id INTEGER NOT NULL,
        product_name VARCHAR NOT NULL,
        price INTEGER NOT NULL,
        quantity INTEGER NOT NULL,
        unit VARCHAR NOT NULL,
        category_id INTEGER,
        PRIMARY KEY (product_id),
        FOREIGN KEY(category_id) REFERENCES categories (category_id)
);
User table:
CREATE TABLE user (
        user_id INTEGER NOT NULL,
        email VARCHAR NOT NULL,
        first_name VARCHAR(150),
        password VARCHAR(150) NOT NULL,
        PRIMARY KEY (user_id),
        UNIQUE (email)
);
```

The database was designed this way to as to have a one to many relationship between categories and product and to have separate cart table for each user and also to help interact with the available categories and products in the database at the home page.

API Design

Used flask API that supports login and crud operations to database.

Architecture and Features

The project follows Flask's framework principles, with controllers situated in the main script for route handling. Templates, utilizing Jinja2, are housed in the "templates" directory, dictating HTML layout and dynamic data placeholders. Models, constructed using Flask-SQLAlchemy, define database structure as classes. Static assets like CSS and images reside in the "static" directory. This architecture ensures a clear organization, simplifying maintenance and scaling while aligning with Flask's design philosophy.

Project: Secure user reg/login (Werkzeug, Flask-Login), Flask-SQLAlchemy DB, dynamic Jinja2 templates, user roles, admin routes, organized static assets - cohesive, efficient Flask app.

Video

https://drive.google.com/file/d/1fEVa8xmYIwJ7xttNKS43fSjXylDbbL_W/view?usp=sharing