

Author

RADHAKRISHNAN B

A passionate B.E CSE student also pursuing an IITM online BS degree in Programming and Data Science.

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 products. Thus, it is used in creating a platform between the seller (admin) and the customer (user) 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 smallscale applications.
7. **JAVASCRIPT:** It is used in HTML files in <script> tags for various functionalities to work properly and efficiently.
8. **Bootstrap:** Bootstrap is a popular open-source CSS framework that provides pre-designed HTML, CSS, and JavaScript components for building responsive and mobile-first web applications. It simplifies the process of creating a consistent and visually appealing user interface by offering a set of customizable and reusable UI components.
9. **CSS:** CSS is a fundamental technology used for styling web pages and defining the presentation of HTML documents. It allows developers to control the layout, colors, fonts, and spacing of elements on a webpage.

DB Schema Design

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

Table

cart

cart_item

category

product

user

3 rows

Actions	id ▲ ▼	username ▲ ▼	password ▲ ▼	is_admin ▲ ▼
	1	rk123	pbkdf2:sha256:600000\$3JyecgskHb2bY3jX\$0dc2cf2d9cb79eb750cddd051d3a10bf98248a696158a5fe472af5ea1db31e12	1
	2	User1	pbkdf2:sha256:600000\$CuZW8FTNu6qk3YbA\$3f9c13fd6d5b0eb25f2cf35eec6f53117b37be5acbd82489d7c74d6e6f124573	0
	3	User2	pbkdf2:sha256:600000\$uRAJYNTt7rVqhYGaF\$46d64bb6d20e42c769874693906ac94cb463c73d6531e3e7b3bab1e8ffb897b1	0

Actions	id ▲ ▼	name ▲ ▼	price ▲ ▼	quantity ▲ ▼	category_id ▲ ▼
	1	Potato	30	20	1
	2	Peas	170	15	1
	3	PineApple	60	1	2
	4	Bean	17	95	1
	5	Papaya	52	30	2
	6	Banana	48	22	2
	7	Kesari Rava	60	50	3
	8	Rava Idli	60	35	3
	9	Rice Flour	48	50	3
	10	Atta	75	60	3

Actions	id ▲ ▼	name ▲ ▼
	1	Vegetable
	2	Fruits
	3	Flour

Actions	id ▲ ▼	user_id ▲ ▼
	1	2
	2	1

Actions	id ▲ ▼	product_id ▲ ▼	quantity ▲ ▼	cart_id ▲ ▼
	1	1	3	2

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 registration/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/1M0B1Z7CEb9BTfx8djEpWFTbHlyTqbeXj/view?usp=drive_link