Author

Name: Ayan Nayyer
Email: 22f3000961@ds.study.iitm.ac.in
ID: 22f3000961
About me: I'm currently pursuing BS in Data Science and Applications as my standalone degree and currently in Diploma level. I have gained a lot of coding skills and insights of HTML,Python,Flask,Jinja etc. upon working on this Application Development Project

Description:

Developed a Flask-based web application for a grocery store (BasketIT). Implemented CRUD functionalities for products and sections, a cart system for user purchases, and APIs to fetch sections and products data for display.

Technologies Used:

Python: Backend logic development for its readability and versatility.
HTML: To develop the Web Pages.
CSS: To style to Web pages.
Flask: Web framework for routing, rendering templates, and handling requests.
Flask-Restful: To develop the Restful API for the app.
Flask-SQLAlchemy: Simplified database interaction with ORM.
Flask-CORS: To enable CORS for the app.
Bootstrap: For clean and visually appealing UI.
Jinja2: Dynamically rendered HTML templates with Python logic.
SQLite: Easy-to-use SQL database engine for data storage.
Swagger OpenAPI: To create documentation for the API developed for the app.

Database Schema Design:

The database has five tables and the Schemas are as follows:

User Table (user info)	**Section Table (grocery section)**	**Product Table (product details)**
ID (int): Primary key, Auto Increment **Name** (string): unique, not null **Password**(string): unique, not null **Email**(string): unique, not null **is_admin**(boolean)	**ID** (int): Primary key, Auto Increment **Name** (string): unique, not null	**ID** (int): Primary key, Auto Increment **Name** (string): unique, not null **manufacture_date**(date): not null **expiry_date**(date): not null **Rate**(float): not null **Stock**(int): not null **Section_id**(int): Foreign Key(section.id) **Owner**(int): Foreign key(user.id)
Cart Table (Cart items)	**PromoCode**	
ID (int): Primary key, Auto Increment **userid**(int):Foreign key(user.id) **productid**(int):ForeignKey(product.id) **quantity**(int): not null	**ID** (int): Primary key, Auto Increment **code**(string):unique, not null discout_percentage(float):not null	

Constraints:
Primary keys ensure unique identification of records in each table.
Foreign keys establish relationships between tables, ensuring data integrity.
Unique constraint on the promo code ensures each promo code is unique, preventing duplicates.

Reasoning:
The structure efficiently manages grocery store data with relationships between users, sections, products, and shopping carts.
The use of foreign keys helps maintain referential integrity and ensures products and cart items are linked to specific users.

API Design

CRUD on Sections:
Endpoint: /api/sections, /api/sections/{id}
Description: This API allows performing CRUD operations on sections. It supports HTTP methods like GET, POST, PUT, and DELETE to manage section data in the database.

CRUD on Products:
Endpoint: /api/products, /api/products/{id}
Description: This API allows performing CRUD operations on products.

Additional APIs for Getting Sections/Products to Display:
Endpoint: /api/sections/display and /api/products/display
Description: These APIs are used to fetch sections and products data to display on the frontend.

These APIs were implemented using Flask and Flask-Restful to define routes and resources. SQLAlchemy was used to interact with the database and the data is returned as JSON responses by Flask's jsonify. For more information refer to BasketIT.yaml file.

Architecture and Features

The application follows the MVC structure: The models, controller could be found in the application folder whereas the app.py could be found in the root directory. All the html templates could be found in the templates folder. The view of the application is created using HTML,CSS and Bootstrap. Controller is created using Python and Flask. The Model is created using Sqlite.

Project Features Implemented:

- User Authentication: The project includes user registration and login functionality using hashed passwords.
- Manager/Admin Privileges: Certain users can be designated as admin users, granting them additional privileges such as managing sections and products.
- Sections and Products Management: Admin users can add, edit, and delete sections and products. Each product is associated with a section, allowing better organization and management.
- Product Search: Users can search for products based on their name, rate, stock, manufacture date, and expiry date, Implemented using SQLAlchemy's filter methods.

- Cart Management: Users can add products to their cart, specifying the quantity they wish to purchase. Cart contents are stored in the database, and the total amount is calculated based on the rate and quantity. Once they buy, stock updates in the database.
- Out of Stock Notification: System displays an "Out of Stock" message for products with a stock of 0 or sold-out products.

Additional Features:

- Aesthetic Cart Layout, Promo Codes etc.

Overall, the project implements core features for managing sections, products, and user carts, ensuring a smooth shopping experience. Additional features enhance the aesthetics and functionality of the web application, creating an efficient and user-friendly grocery store management system.

Video

For viewing the video, click here!