Author

Ankit Kumar

21f1000513

21f1000513@ds.study.iitm.ac.in

Hi ! I am Ankit , a Data Science enthusiast and also have a keen interest in web development . I love building cool projects and exploring new technologies. This project is result of my efforts and skills to learn and grow.

Description

In this project , the goal is to create a based multi -user grocery web application which will allow user to sign up , login and browse different grocery sections , view products , add them to cart with also an admin interface to manage product and sections.

Technologies used

1. Flask - render_template , request, redirect, url_for.
2. SQLAlchemy
3. Flask–Login - login_required, current_user, logout_user, login_user , UserMixin
4. Flask-WTF - StringField, PasswordField, BooleanField, SubmitField, TextAreaField, IntegerField, FloatField , DateField
5. Datetime – date , datetime
6. Bootstrap
7. SQLite
8. Jinja

In Flask, the Flask-Login extension provides user session management and authentication features. It helps with tasks such as user login, logout, session management, and identifying the current user.

DB Schema Design

- The Grocery web application database is designed with four main tables: Users, Sections, Products, and Cart.
- The Users table stores user information, including the id, usernames, email addresses, passwords and has an is_admin section to indicate whether the user is an admin or not.
- The Sections table contains id , name of the section.
- The Products table contains details about individual products, such as name, price, manufacture date, expiry date, available units, and their association with specific sections.
- The Cart table is used to store information about items added to a user's cart, including the user ID, product ID, and quantity.

Constraints

- Primary keys (id) ensures each record has a unique identifier.
- Unique constraints on username and email columns in the Users table to prevent duplicate usernames and emails.
- Foreign key constraints on section_id, user_id, and product_id columns to ensure referential integrity and enforce relationships between tables.
- Non-null constraints (nullable=False) on critical columns like username, email, password, name, price, manufacture_date, exp_date, section_id, units, user_id, product_id, and quantity to ensure essential information is always provided.

Architecture and Features

- admin.py contains all the controllers related to admin wherein user.py , one will find all those controllers that handles user's request.
- The view functions related to the admin such as homepage of admin, edit section , add section , delete section , edit product etc are present in admin.py whereas view functions related to user such as view products , view cart , delete cart , search products etc are present in user.py .
- grocery.py renders the homepage of the application.
- model.py contains the model definitions of the database.
- form.py contains the definition of loginform and signupform .
- The templates directory contains HTML templates that define the structure and layout of the web pages to be rendered.

Key features implemented in the application

- User Registration and Authentication (using Flask-Login)
- User Roles and Authorization: Flask-Login's current_user object is used to check user roles and authorize access to specific pages.
- Product Sections and Products Management : Administrators can add, edit, or delete product sections as needed and the user can view the sections , products , add them to the cart and delete it.
- Search and Filter Products : Products can be filtered based on availability and other criteria.
- Shopping Cart : The cart keeps track of the total price of the selected items.
- Inventory Management : Product units are tracked to ensure accurate inventory management . If a product is out of stock, users cannot add it to the cart.

Video

Click