Author

Name: Ashifa Muhammedali
Roll Number: **22dp1000014**
Student email: 22dp1000014@ds.study.iitm.ac.in

I am Ashifa Muhammedali, a former primary school teacher. I have completed BSc Biotechnology in 2009 and currently pursuing diploma in programming from IIT Madras

Description:

Leaf Grocery Shop is an online platform for purchasing various products. Users can easily browse and buy items across different categories. Manager and Admin can manage products and categories, with manager-initiated category changes requiring admin approval. This ensures an organised structure and enhances the user experience.

Technology:

1. Flask: Python-based web framework for building web applications.
2. Flask-restful: simplifies REST API creation within flask application
3. Flask-SQLAlchemy: Integrate SQLAlchemy for database interaction in flask, simplifying database operations using python object
4. Flask-Security: Token based Authentication
5. Vue.js: Lightweight JavaScript framework for building interactive web interfaces and single page applications.
6. Bootstrap: Front-end framework for creating responsive web design.
7. DB Browser for SQLite: Visual tool for managing SQLite database efficiently.
8. SQLite3: Lightweight, serverless database engine for embedded and small-scale applications.
9. Smtplib: Enables application to send emails by connecting to SMTP servers for automated email transmission
10. Celery: Manages scheduled and user-triggered tasks, allowing background job execution and user alerts.
11. Redis: Acts as message broker and result backend for celery, managing tasks queuing, distribution, and result storage. Additionally, Redis for optimizes data retrieval and storage by providing caching capabilities within the application.
12. MailHog: Testing tool that capture outgoing emails, enabling developers to test email features without sending to real recipients.

DB Schema Design:

1) **user table:**
   Attribute: id (primary key), email, username, password, active,fs_uniquifier,timestamp
2) **role table**
   Attribute: id (primary key), name, description
3) **roles_users table:**
   Attribute: id (primary key), user_id (foreign key), role_id (foreign key)
4) **category table:**
   Attribute: category_id (primary key), category_name, imagelink, approve, updateRequest, deleteRequest
5) **product table:**
   Attribute: product_id (primary key), product_name, Description, Category_id (foreign key), price_per_unit, quantity, Stock, image_url, manufacture_date, expairy_date, timestamp
6) **cart table:**
   Attribute: cart_id (primary key), user_id (foreign key), product_id(foreign key), price_per_unit, quantity, total_price
7) **order table:**
   Attribute: order_id (primary key), user_id (foreign key), total_price, order_date, payment_id (foreign key), address_id (foreign key)
8) **order_item table:**

Attribute: order_item_id (primary key), order_id (foreign key), product_id (foreign key), price_per_unit, quantity, total_price

9 **payment table:**
Attribute: payment_id (primary key), user_id (foreign key), type, card_number, cvv, expiry_date

10) **address table:**
Attribute: address_id (primary key) ,user_id (foreign key), street, city, state, postal_code

Relationships:

many to many relationships:

1. Role and user: Multiple users <--> multiple roles

one to many relationships:

1. User and orders: single user -> multiple orders
2. Category and product: each category -> Multiple products
3. Product and carts: single cart -> Multiple products
4. cart and payment: single cart -> Multiple payments options
5. cart and address: single cart -> Multiple address options
6. order and order_item: one order -> Multiple order_item

one to one relationship:

1. order and payment: single order <-> single payment
2. order and address: single order <-> single address

Architecture and features:

Flask Backend and Vue.js Frontend: Leaf Grocery Shop integrate Flask for the backend infrastructure and vue.js for a dynamic frontend experience

Backend Functionality: flask manages core operations such as user authentication, product, category managements, cart handling, order processing

Frontend User Interface: Vue.js powers an engaging and responsive user interface, ensuring a seamless browsing and purchasing experience

Integration: the flask backend seamlessly communicated with vue.js frontend components, enabling real time data exchange and delivering an integrated user experience

File structure overview:

Root file name is **mad2_project**. This folder consists of following files,

1) instance/database.db: stores application data
2) static/components: contains all the components essential for frontend interface
3) template/index.html: primary entry point for application frontend.
4) app.py: main entry point for the backend server and application logic
5) requirements.txt: all the packages need for the application
6) application.py: this file contains all the backend logic and backend job
7) static/app.js: contains Vue instance and all the routes for frontend navigation

key functionality and features:

The Leaf Grocery shop includes essential features such as:

1) seamless browsing and purchase of products across diverse categories.
2) Manager and Admin roles for product and category management, requiring Admin approval for manager-initiated category changes
3) Secure user authentication and authorization, ensuring personalised shopping experience
4) Efficient cart operation, order placement and payment handling

Video:

https://drive.google.com/file/d/1U63Bh9D3oMQby8cfn6qwPOWGLkorcZJP/view?usp=sharing