Project Report: Flask Grocery Store V2

Abdul Ahad Rauf
21f3002590
21f3002590@ds.study.iitm.ac.in
8948887405

ABSTRACT:

The Flask Grocery Store Application is a dynamic web-based platform that offers a seamless online grocery shopping experience. Leveraging the Flask framework, it allows users to easily browse, search, and purchase a wide range of grocery items. The application features robust database management using SQL_Alchemy, secure password encryption and integrates various technologies to replicate the functionalities of a physical grocery store in a virtual space. It's a comprehensive solution, blending e-commerce efficiency with user-friendly design, to create a convenient and reliable online grocery shopping environment.

TABLE OF CONTENTS:

1.INTRODUCTION:

I, Abdul Ahad Rauf; am pursuing BS Degree in Data Science and Programming from Indian Institute of Technology, Madras. I like to code and I have made this project for my MAD-2project. Flask Grocery Store Application aims to provide a user-friendly platform for online grocery shopping. Users can conveniently browse through various grocery items, add them to their cart, and place orders from the comfort of their homes. The project is built using the Flask web framework, allowing for efficient development and deployment of the application.

Video=>
https://drive.google.com/file/d/1TuhotGZjUWok0omXUivNnKjQzj05wK2H/view?usp=sharing

Api Docs=>

https://drive.google.com/file/d/1QrTwg3xrP8-6w1JmEAKwrXFoUC4xkJNg/view?usp=sharing

2.TECHNOLOGIES USED:

Flask: Web framework for building the application.
SQLAlchemy: Database ORM for managing data.
Flask-Login: User authentication and session management.
Werkzeug: For password hashing.
Flask_caching: For cacheing.
SQLite: Database management system.
JavaScript VueJS: Front-end development (Bootstrap).
Matplotlib: Data visualization for analysis.
Redis & celery : For backend jobs
Fpdf and Weasyperint: for pdf and html creation

3.FEATURES AND FUNCTIONALITIES:

User Registration and Login , Logout

Product Browsing and Searching

Cart Management

Order Placement and History

Admin Dashboard for Product Management

Monthly and Daily Remainders

PDF generation for Cart items.

Store analysis for admin.

CATEGORY AND PRODUCT MANAGEMENT

Grocery-Store-V2 enables efficient management of product categories and individual products. store managers, can perform CRUD (by sending a change request to admin)for operations on categories as well as products, ensuring a well-organized and up-to-date inventory.

Request Handling

The system facilitates the creation, viewing, and deletion of user requests, streamlining the communication between store managers and administrators.

API Functionality

A well-defined RESTful API provides extensive functionality for posts, users, comments, and follows, promoting seamless integration with external applications.

Analysis

A well defined report creation for users and store manager and admin

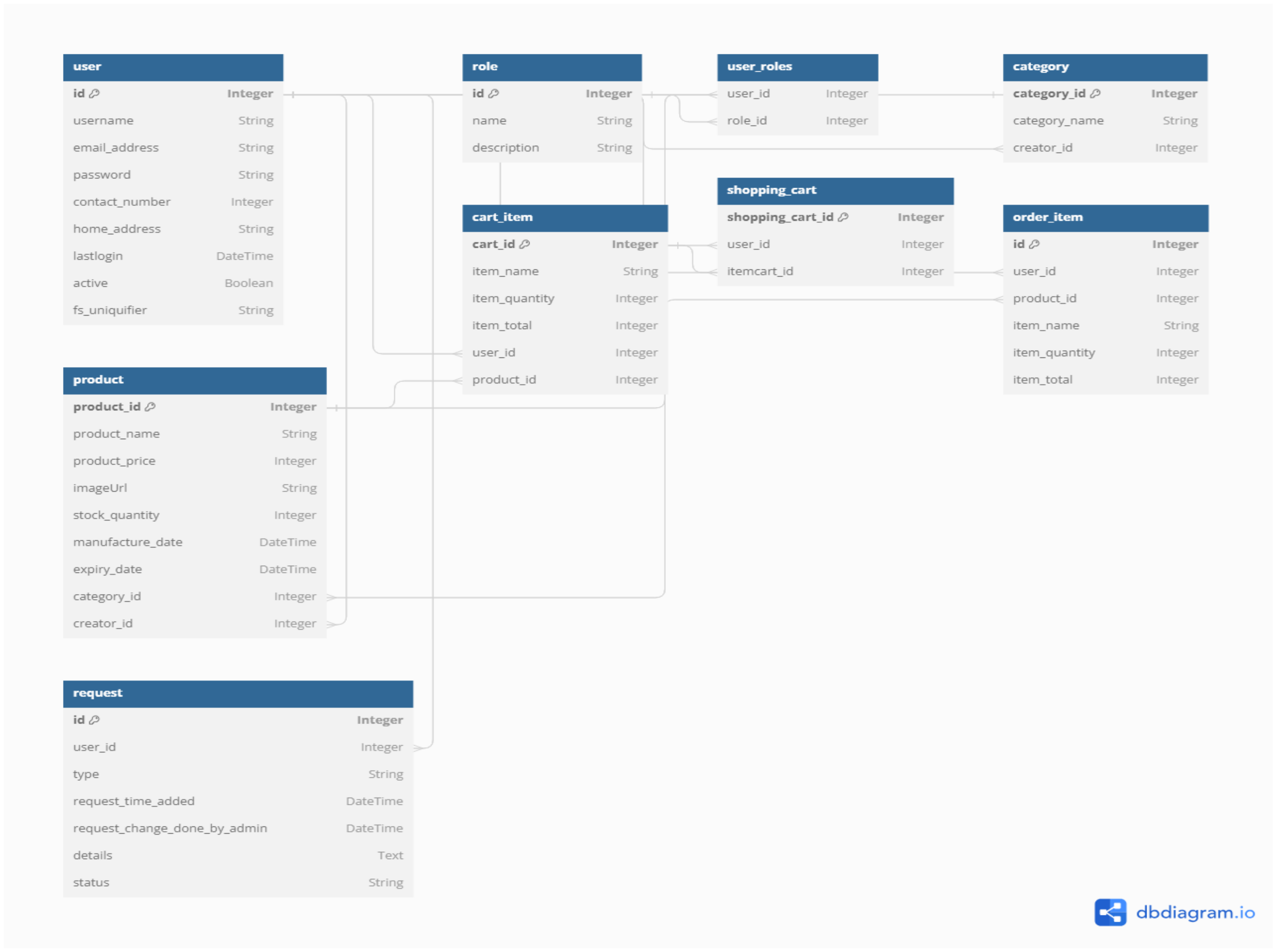
Mailing or Download as PDF , CSV , HTML

Reports can be downloaded in any format the user wants

4.Video Link:

https://drive.google.com/file/d/1TuhotGZjUWok0omXUivNnKjQzj05wK2H/view?usp=drive_link

5.System Architecture:



user	
id	Integer
username	String
email_address	String
password	String
contact_number	Integer
home_address	String
lastlogin	DateTime
active	Boolean
fs_uniquifier	String

role	
id	Integer
name	String
description	String

user_roles	
user_id	Integer
role_id	Integer

category	
category_id	Integer
category_name	String
creator_id	Integer

shopping_cart	
shopping_cart_id	Integer
user_id	Integer
itemcart_id	Integer

cart_item	
cart_id	Integer
item_name	String
item_quantity	Integer
item_total	Integer
user_id	Integer
product_id	Integer

order_item	
id	Integer
user_id	Integer
product_id	Integer
item_name	String
item_quantity	Integer
item_total	Integer

product	
product_id	Integer
product_name	String
product_price	Integer
imageUrl	String
stock_quantity	Integer
manufacture_date	DateTime
expiry_date	DateTime
category_id	Integer
creator_id	Integer

request	
id	Integer
user_id	Integer
type	String
request_time_added	DateTime
request_change_done_by_admin	DateTime
details	Text
status	String

6.More Details:

The Flask Grocery Store Application is an advanced web-based platform, intricately designed to offer a comprehensive online grocery shopping experience. This application stands out with its integration of several cutting-edge technologies and frameworks, each contributing to its robust functionality and user-friendly interface.

Flask Framework: The core of the application is built using Flask, a lightweight and versatile web framework in Python. Flask provides the backbone for routing, request handling, and other web interface functionalities, ensuring a smooth and responsive user experience.

Flask-Security and Role-Based Authentication: Security is paramount in this application. It employs Flask-Security, an extension that facilitates secure user authentication and authorization. The system incorporates role-based authentication, where users are assigned different roles (e.g., admin, shopper) with varying access levels, enhancing security and operational efficiency.

Flask-RESTful for API Management: To handle API requests efficiently, the application uses Flask-RESTful. This extension simplifies the creation and management of REST APIs, enabling seamless integration and communication between the frontend and backend.

Token-Based Authentication : Enhancing security further, the application uses token-based authentication. This method ensures that each session is securely authenticated, protecting user data and preventing unauthorized access.

Celery and Redis for Task Scheduling and Reminders:For efficient task management and scheduling, such as sending reminders or processing background tasks, the application utilizes Celery with Redis as the message broker. This combination ensures timely execution of tasks without overloading the main application server.

MailHog for Email Handling: MailHog is integrated for handling email interactions, such as order confirmations and password resets. It's a lightweight tool, ideal for development and testing email functionalities in the application.

-WeasyPrint for PDF Generation: For generating PDFs, such as receipts or reports, WeasyPrint is used. It's a powerful tool that converts HTML/CSS documents to PDF, ensuring high-quality and accurate document rendering.

-Vue.js for Frontend Development: The frontend of the application is developed using Vue.js, a progressive JavaScript framework known for its simplicity and flexibility. Vue.js enhances the user interface with dynamic, reactive components, making the shopping experience more engaging and intuitive.

7.API DOCUMENTATION:

This section provides comprehensive documentation for the Flask Grocery Store Application's APIs, detailing all CRUD (Create, Read, Update, Delete) operations related to users, carts, products, categories, and the checkout process. The documentation is meticulously organized in a YAML file, ensuring ease of use and clarity. It includes specifics on API endpoints, request/response formats, authentication methods, and error handling. This thorough documentation serves as a valuable resource for developers and integrators, offering clear guidelines for interacting with the application's diverse functionalities.

See Api Documentation :
https://drive.google.com/file/d/1QrTwg3xrP8-6w1JmEAKwrXFoUC4xkJNg/view?usp=sharing

8.References:

1.Flask Framework:
https://flask.palletsprojects.com/

2.Flask-Security:
https://github.com/mattupstate/flask-security

3.Flask-RESTful:
https://flask-restful.readthedocs.io/

4.Token-Based Authentication:
Flask-Security-Too
5.Celery:
https://docs.celeryproject.org/en/stable/

6.Redis:
https://redis.io/documentation

7.MailHog:
https://github.com/mailhog/MailHog

8.WeasyPrint:
https://weasyprint.org/documentation

9.Vue.js:
https://vuejs.org/