Mad 2 Project Report Sept Term - 2023

Author: - Amit Kumar

Roll No: - 21f3002445

Email: - 21f3002445@ds.study.iitm.ac.in

About me:

Hey, currently I am student of B.S. in Data Science from IIT Madras. I am interested in data science field. Through this course I also have knowledge in website development using Flask and Vue.js framework. I thrive on engaging in competitions and technical hackathons, as they allow me to showcase my skills and learn from other like-minded individuals.

Title Of the Project: Grocery Store V2 Application

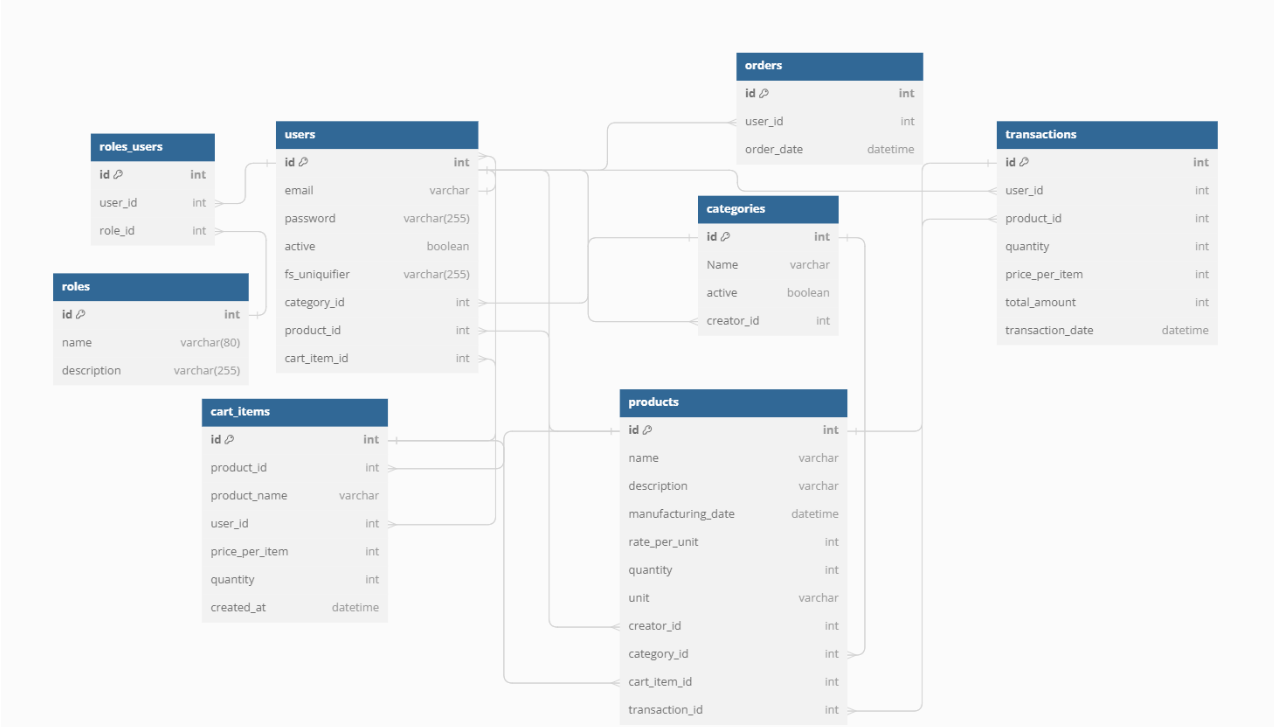
Description:

In this project, my role is developing a robust web application for a Grocery Store using Flask, Vue.js and celery for and integrating database functionalities with SQL Alchemy. I am responsible for creating user and admin, manager authentication systems using RBAC and implementing CRUD operations with Api for categories and products, ensuring data integrity, designing responsive front-end interfaces and style through using CSS.

Technologies used:

1. Flask: flexible framework for building Web applications.
2. Vue.JS: JavaScript framework that facilitates the UI (user interface) development of websites and single-page applications. A progressive JavaScript framework, Vue makes creating user interfaces simpler and more enjoyable.
3. Flask -SQL Alchemy: these technologies use for making simplifies database interaction provide efficient data handling and ensure the data integrity.
4. Flask-Security: this technology helps in for user authentication, authorization, role-based access and Token generation, and session management.it is also help in find out the current user.
5. Flask-Restful: this technology helps in creating API for applications.
6. Celery: It is used for performing backend job for sending the periodic task and asynchronous work.
7. Redis: it is used for offers a fast, in-memory data store to power live streaming use cases. Redis can be used to store metadata about users' profiles and viewing histories, authentication information/tokens for millions of users,
8. Redis for caching: it is used for increased the performance for applications.
9. Bootstrap: it is used for frontend design.
10. Html-CSS: for structuring and designing frontend of our website.

DB Schema Design:



Architecture and Features

Root Directory: The root directory contains the main application file Main.py, celeryconfig.py for performing the backend job, a cenfig.py for handle the configuration of app, upload_intial_data.py for creating the initial data in database application folder for handling the backend work, an instance folder for the database file, monthly reports folder for generate monthly wise report ,static folder for performing the frontend work, a template folder and other project-related files like virtual environment test, requirements.txt, and the.

Application Directory: This directory serves as a package and contains the backend core functionality of the application.

Mail_service.py: This directory handles the mail service required all the package and port addresses and connect the Smtp server.

Instances.py: Marks the directory as handling for caching that improve the performance of application.

models.py: Defines the SQL Alchemy models for database tables like User, Role, Category, Product, Cart Item, Order and Transaction.

Resource.py: This directory used for creating the required API for application.

Sec.py: This directory used for stopping the circular import.

Tasks.py: This directory used for generating the schedule task and creating for report.

Worker.py: This directory used for handling the celery and initialising the celery for perform the schedule job.

Static Directory: This directory handles the frontend work.it has component folder, index.js and router.js.

Index.js: It is connecting the Vue.js and route.js.

Router.js: Defines the different routes and their corresponding view functions.

templates/: Contains HTML templates for different views, such as index.html, Monthly_user_report.html.

The project follows a modular structure where each component has its dedicated file or directory, promoting code organization and maintainability. The application directory acts as the core of the application for backend job, containing models, resource, task, and worker assets. Statics directories handle the frontend job.

Features & Functionality:

1. **User login and signup page**: all users are login and role based they are going to their home correspondingly.

2. **User Interfaces:**
   1. **Admin Dashboard**: Provides admin with access to category management. The dashboard offers intuitive forms for adding/editing /deleting category, also able to activate manager, approve the category request from store manager.
   2. **Store Manager Dashboard:** Provides store manager with access to product management. The dashboard offers intuitive forms for adding/editing /deleting product, also able to request category for product to admin, and able to download the pdf.
   3. **User dashboard**: Allows users to browse products, view details, and order the product, cart the product, remove the cart product able to update the profile, search the product on the various parameters.
3. **Monthly Expenditure Report**: Another valuable feature is the generation and delivery of a comprehensive Monthly expenditure Report. This report provides users with insights into their interactions with the platform throughout the previous month. At the beginning of each month, a scheduled job is initiated within the system to generate the Monthly expenditure Report. The HTML-based email report which contains the details of their order and present it in a clear and engaging format. Also, able to download the pdf.
4. **User-Triggered Async Job**: The app empowers store manager with the ability to initiate user-triggered asynchronous jobs for exporting product-related data in CSV format. The exported CSV file includes the following information product name and remaining quantity, product description triggers the asynchronous export job by clicking a designated button. The export job is managed using Celery. Upon successful completion of the export, users receive an alert informing them that their requested CSV file is downloaded.
5. **Implementing Caching for Enhanced API Performance**: caching is used to optimize the API performance and reduce the load on the server and to maintain data accuracy and freshness, cache expiry mechanisms is used. Used Redis, to implement caching effectively.

Video Link:

https://drive.google.com/file/d/18hbd1B9YLQJGp9cIHtQAywXc9r5ZL-hn/view?usp=sharing