**Grocery Store Application**

**A Project report for the MAD2**

Submitted by

Trivedhan Sivaprakash (21F1001724)

21f1001724@ds.study.iitm.ac.in



IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

**Abstract**

This project aims to create a fully functional Grocery Store Application using Python Flask Module, Vue Js and other relevant extension modules that incorporates both RBAC mode of login in with Flask Security Token Based Authorization sign in to create and modify categories as well as shoppers to buy and sell the respective products. The shopper can add the list of products and their respective quantities in the shopping cart and then purchase them while the store owner can approve the order and process it for delivery and payment.

**Modules Used**

The following modules were used for creating and the functioning of this project.

1. Python based Modules
2. Flask 🡪 Overall designing and structuring of the application was done using Flask.
3. Flask Restful 🡪 This module was used to create the API’s needed to perform the CRUD operations in the application.
4. Flask SQLAlchemy 🡪 This module was used for the creation and configuration of the necessary Tables.
5. Flask Security 🡪 RBAC Token based Sign In
6. Celery Jobs and Redis 🡪 for sending mails and cacheing
7. Werkzeug 🡪 This module was predominantly used to trigger error validations when needed.
8. Requests 🡪 This module was fundamentally used in the calling the designed API’s when an endpoint was launched.
9. Front End Development
10. Vue Js (For Structuring and UI Designing of the Application)]
11. Bootstrap (Screen Enhancement)

**Back End Database Structure**

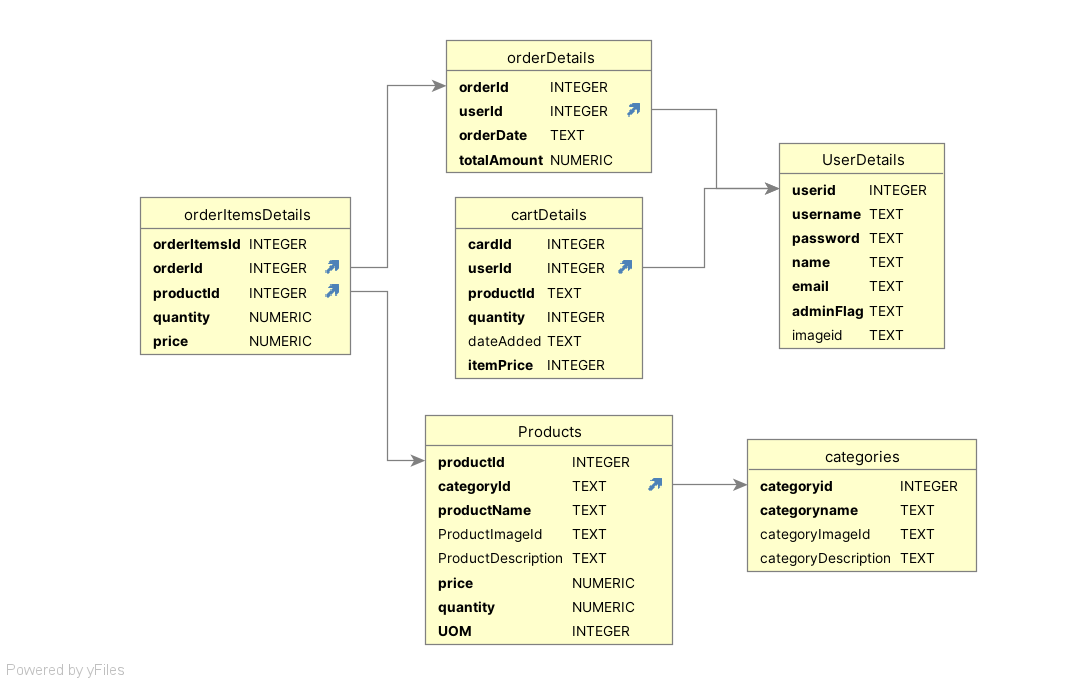
****

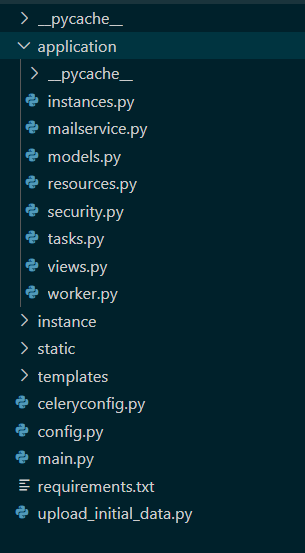
Figure 1: Database Structure

The project utilizes six tables for its overall functionality. The User, Role and Role\_User tables stores the pertinent details regarding the user in accordance with the RBAC Method. This table is the parent of the tables OrderDetails and Cart Details which stores the Order ID associated with the sure and the respective products purchased. The orderItemsDetails Table stores the list of orders associated with the order. The Product tables has a list of all the products along with the stock availability. This is linked to the Categories table which stores the list of all the categories available.

**API Design**

The API’s form the integral backbone for this application. They help in transferring of the data from the application interface and the database. This application implements 12 API’s for fetching the data to the screen, updating entered data and deleting selected data from the screen. Other than Sign-in, all the other screen’s implement API’s for fetching and the transfer of data. The list of API’s and their respective endpoints and uses can be found in the YAML file found in the project directory.

**Architecture and Features**

The project directory comprises of four folders – application (This comprises of all the relevant python codes with configuration details, controllers, API structures, models, resources, views, Database structures), instance (This contains the SQLLite Database used for backend storage), static (All the JS Vue Jx ), templates (This contains all the HTML templates used for the project) and driver code main.py (Which needs to be executed in the command line for the application to run). The structure mentioned is illustrated in the image provided.

The application has measures to create own user details and use it for login to differentiate between multiple users. The first user in the database however is an admin user and the admin user had the permission to create new categories and each of its respective set of products. The admin user has a dashboard that can be used for viewing the total sales revenue gathered and the list of orders ordered in the application and the total shoppers available. The shoppers can view the products logged by the admin user and add these products to their unique list of carts. After addition, they can then select the list of products they require and then purchase those products.

Figure 2: Project Organization Structure

**Video**

https://drive.google.com/file/d/1v1SQh7ibpR\_7ntA-X0QAn83W-l9tMURV/view?usp=drive\_link