Grocery Store– Report

Author

Name: Lalit Kumar

Roll No: 21f3003123

Email: 21f3003123@ds.study.iitm.ac.in

About Me: I am from Barabanki, Uttar Pradesh, and doing B.S. in Data Science and Applications at IIT Madras. Currently, I am in the Diploma level of this program, working diligently to enhance my knowledge and skills in this exciting field.

Description

The grocery store is a multi-user web application that allows users to explore and buy listed products. Similarly, sellers can register and list their products on it. The Rest APIs on products and categories are also developed and follows the OpenAPI specifications.

Technologies used

Python – Python is the primary programming language used for developing app

HTML/CSS – For creating web pages and styling them

Jinja2 – Used for displaying data dynamically in HTML pages

Bootstrap – It is a CSS framework, we used for making pages visually appealing

SQLite- Serves as the database for the application

Python Flask – Serves as the web framework for the app, it utilizes the following libraries.

- Flask-Restful – Used for developing Restful APIs
- Flask-SQLAlchemy – For creating models that allows application to interact with database
- Flask-CORS – Used for allowing cross-origin requests for APIs
- Flask-Login – Used for implementing login functionality to the application
- Flask-JWT-Extended – Used for implementing token-based authentication for API end points

Matplotlib- To create charts of the most demanded products and categories

DB Schema Design

The database mainly has 6 tables including two search tables -

User	Category
user_id (Integer) – Primary Key, Auto Increment **username** (String) – Unique, Not Null **password** (String) – Not Null **role** (String) – Not Null	**category_id**(Integer) – Primary Key, Auto Increment **category_name** (String) – Unique, Not Null

Product	Cart
product_id (Integer) – Primary Key, Auto Increment **product_name** (String) – Unique, Not Null **product_description** (String) – Not Null **unit** (String) – Not Null **price** (Float) – Not Null **quantity** (Integer) – Not Null **mfg_date** (Date) – Not Null **expiry_date** (Date) – Not Null **product_image** (String) – Not Null **category_id** (Integer) – Not Null, Foreign Key(Category.category_id)	**user_id** (Integer) – Not Null, Foreign Key(User.user_id) **product_id** (Integer) – Not Null, Foreign Key(Product.product_id) **item_quantity** (Integer) – Not Null

CategorySearch	ProductSearch
category_id (Integer) – Primary Key, Auto Increment **category_name** (String) – Unique, Not Null	**product_id** (Integer) – Primary Key, Auto Increment **product_name** (String) – Unique, Not Null **product_description** (String) – Not Null

API Design

The APIs are designed using Flask-Restful for performing CRUD on product, category, and user tables. We used the flask-jwt-extended package for implementing authentication and authorization on API endpoints. For more details, please refer to the openapi.yaml file.

Architecture and Features

The application follows the standard MVC architecture. The View of the application is created using HTML/CSS, and Bootstrap. The Controller is created using Python and Flask. Models are the Python classes that map with SQLite database tables using the Flask-SQLAlchemy ORM library.

The features of the application are as follows-

- Signup and Login for User and Admin
- An admin can view, create, update, and delete a product or category
- A user can view, add, increase/ decrease the item quantity, or delete a product from the cart. Users can search for a product or category and add it to the cart
- Search feature is implemented using FTS5 and allow searching by product name, product description, or category name
- APIs for performing CRUD on product, category, and user tables. The endpoints are protected using JSON Web Token
- Frontend as well as backend validation on APIs of input fields before storing/ selecting from the database

Video

For the video link, click here.