Author

Rohit Gupta
21f3001404
21f3001404@ds.study.iitm.ac.in

Apart from this BS Degree, I am pursuing B.tech. In Information Technology Branch from AKTU university Lucknow. I have been into software development since my 1st year of college. And I am looking forward to being a software engineer first and then shift to data scientist.

Description

I have to develop a web application for a grocery store. This online store has one manager and multiple users(buyers). The manager can manage the products on the grocery store portal and users can buy the products from there. Both manager and user have their features and functionality.

Technologies used

Flask: Used to send the response for the server and complete routing
Flask-SQLAlchemy: Used for using SQLAlchemy with flask
SQLAlchemy: User for using SQL in Python and creating object schemas
Jinja2: Used to show dynamic content in the HTML templates
Matplotlib: To create stats graphs of the sales in summary

DB Schema Design

- Manager *(id, manager-name, password)*: To make sure the manager is logging in
- User *(id, user-name, password, address, phone)*: To log in user and sore delivery address and contact details
- Product (*id, name, price, unit, manufacture date, expiry date, quantity, category-id*): Store the product details in all the fields and to know the category it also stores category-id, if a product is out of stock user cannot buy that product.
- Category (*id, name*): Just store the category name and nothing else, each category will have a unique name and that can be changed further.
- **Cart (*id, product-id, user-id, product-name, quantity*):** It stores each cart entry of every user and a user can fetch their cart just by fetching the cart entries of their user-id, this makes it easy to manage the cart in the database tables
- BuyLog *(id, product-id, product-name, user-id, quantity, category-id, category-name, date, time)*: Store each information about every product sold. Users can see their order history and the manager can see the complete history of each product sold.

API Design

Mostly I have not used a dedicated API, everything is happening in the Routes. And the routes are mentioned in the yaml file that I will be uploading.

Architecture and Features

The complete project is nicely structured

- All Routes
    - User Routes: *all the user related routing and api*
    - Manager Routes: all the manager related routing and api
- Database
    - Config DB : *configure details of the database*
    - DB Controller: *controls all the operations happening on the database*
    - Models: All the object schemas are
- Static
    - *Images used in the application*
- Templates
    - Manager Templates: *all pages linked to the manager*
    - User Templates: *all pages linked to the users*
- App.py: *first file run in the whole application*
- Config.py: *configure the application*
- Routes.py: *contains the imports of the routes in the all routes folder*

Video

https://google.com