Author

Kaustubh Singh 21f3002304

21f3002304@ds.study iitm.ac.in

I’m Kaustubh Singh, currently I am pursuing a B.Tech in ECE along with this BS in IITM course. Currently. I am in my Diploma level of the program.

Description

We had to design an app which could register/login users, let them select and search products from a variety of categories and give the ability for store managers to add/remove/edit products as well as categories of the products.

Technologies used

flask – used to make the actual app and make it connect to the network.

SQLAlchemy – To make the ORM which helps us connect with the sqlite3 database.

Flask-SQLAlchemy – To help integrate SQLAlchemy with the Flask app.

Flask-restful – To implement the API.

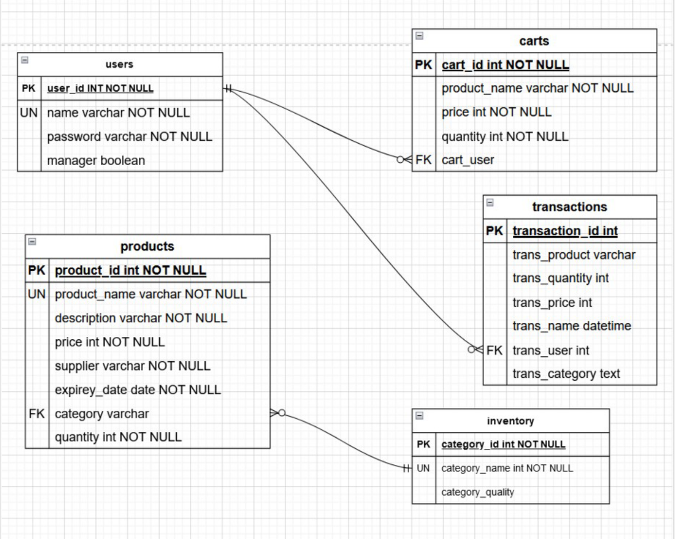
Flask-cors – To help cross origin requests to the API.

Pillow – To help convert binary string data to JPEGs

Matplotlib - To help plot the data

Pandas - To read the data and convert it for Matplotlib.

DB Schema Design



The first important thing to note is how the products and inventory are not connected to the user. This was done to prevent any changes made to the inventory or products drastically affecting any user information and vice versa. The inventory can be given other categories later to affect the entire product lineup such as discounts. The products table stores all information about the products. The user is connected to two tables, cart and transactions. The cart stores the information of the current list of products the user is buying. After the transaction is done, the cart information is stored in the transaction table. The name of the transaction is the time at which the transaction was made so you can look up a unique transaction with transaction name and transaction user.

Architecture and Features

The project is organised into three folders at first. .venv, instance and store. The instance folder has the database which the project generates through the 'db.create_all()' command. It is used to store all sensitive data of the program, but in this case we are using it to store the database. The virtual environment stores all the python packages and finally we have 'store' which has all the main files of the app. The app then has the static folder where all the static files like the transaction record, photos and css files are stored. Then there is the templates folder which stores all the html files and templates required. Then the app has the __init__.py file which is where the main Flask app is defined. In this project I'm using Blueprints which is a feature of jinja because it helps me maintain clarity of code and keeps the file from cluttering and becoming confusing. So the file auth.py handels authentication such as login,logout,register features. Manager.py handels all the manager related tasks of the app.browse.py is the part which shows the main app and allows us to browse it. Cart.py handels all the cart related parts such as adding to cart, checking out and storing it to memory. stats.py handels the part about showing the statistics of the transactions. Model.py has our database defined to the SQLAlchemy ORM. API.py is the file which handles all the API requests. API_validate.py has the functions and objects needed for API.py.

There is every basic functionality in this app such as searching which was implemented by using sqlite3 fts5 on the products table. Every user can individually cart products and buy them. The store manager can add categories, remove them or edit them. The store manager can also add products, delete products or edit them. If the products are out of stock it is displayed immediately on the screen. On top of this there is an API which lets you have full CRUD functionality on products as well as categories. Then you can also click on the statistics button as the manager to find out about your sales over time.

Video

https://drive.google.com/file/d/1ydhdbyed2h62La4YbvEjVxNCXRIZXxj1/view?usp=sharing