Author:

Naveen Kumar K
22f1000371
22f1000371@ds.study.iitm.ac.in

I’m a final year student pursuing Btech Biotechnology at Muthayammal Engineering College,Rasipuram,Tamilnadu. I love to code and build and develop solutions that could be helpful for a community.

Description:

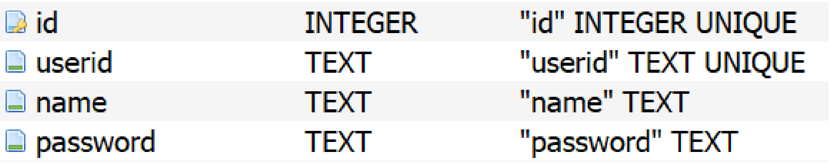
Grocery store is an e-commerce web application. It is used to buy Groceries by user and add, delete and edit products and categories by the Admin. There should be a signup/login page for creating an account and logging into their account. Admin should have a separate login page. Admin should have a category section ,product section and summary. Users should have a search option and cart section to buy products.

Technologies Used:

- HTML - Template
- Flask
- Bootstrap - Stylesheet
- SQLite - Database
- Flask-SQLAlchemy - Connection with database to the app
- Jinja2 - Template rendering
- Matplotlib - Graph plotting for summary

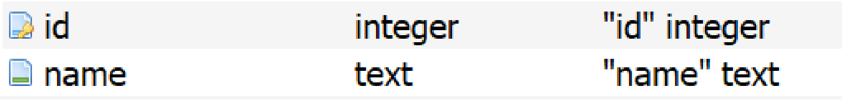
DB Schema Design:

1.User



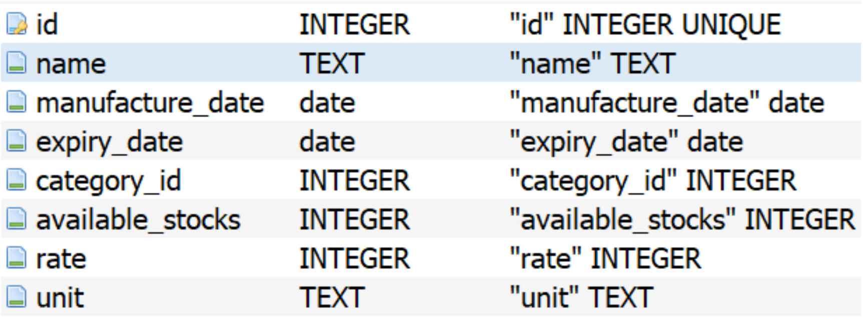
id	INTEGER	"id" INTEGER UNIQUE
userid	TEXT	"userid" TEXT UNIQUE
name	TEXT	"name" TEXT
password	TEXT	"password" TEXT

2.Category



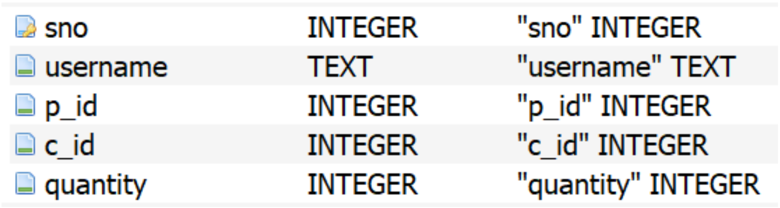
id	integer	"id" integer
name	text	"name" text

3.Product



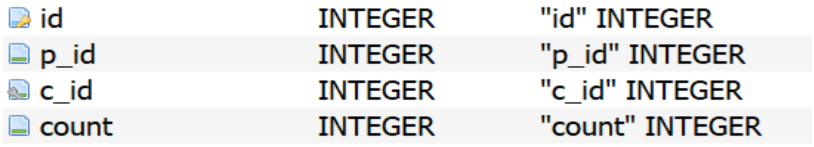
id	INTEGER	"id" INTEGER UNIQUE
name	TEXT	"name" TEXT
manufacture_date	date	"manufacture_date" date
expiry_date	date	"expiry_date" date
category_id	INTEGER	"category_id" INTEGER
available_stocks	INTEGER	"available_stocks" INTEGER
rate	INTEGER	"rate" INTEGER
unit	TEXT	"unit" TEXT

4.Cart



sno	INTEGER	"sno" INTEGER
username	TEXT	"username" TEXT
p_id	INTEGER	"p_id" INTEGER
c_id	INTEGER	"c_id" INTEGER
quantity	INTEGER	"quantity" INTEGER

5.Summary



id	INTEGER	"id" INTEGER
p_id	INTEGER	"p_id" INTEGER
c_id	INTEGER	"c_id" INTEGER
count	INTEGER	"count" INTEGER

API Design:

APIs are designed logically to support the rest of the application. The API part is different from the actual application.

1. User:

CRUD operations on User to get and add users.

2. Category:

a. With id: get, put, delete

b. With id: get (view all the categories created by the admin), post (categories created by admin),delete(remove category)

3. Product:

a. With id: get, put,delete

b. With id: get (view all product of a category), post (create a product under a category by admin),delete(remove product)

4. Cart:

a. With username:get,put,delete

B. With username: get(view all product in cart),post(add product to cart),delete(remove product from cart)

Architecture and Features:

- The app routes are located in the routes folder.
- The app configuration is written in app.py.
- Database schemas are written in grocery.sqlite3.
- The templates are in the templates folder.
- Media content is in the static folder.
- Features:

1. Implementation of creating an account ,logged in user and admin..
2. Usage of Modals for add category, edit category, delete category by admin.
3. Usage of Modals for add product, edit product, delete product by admin.
4. Usage of Models for add product, edit product, Edit product to cart by user
5. Summary page with product sold summary and summary for each product along with a graphical representation with a bar graph.

Video link:

https://drive.google.com/file/d/1D2jNTxzOqqEQoTtkbmCyJrR1uBcv2SQi/view?usp=sharing