Author

Amal Roy

21f1004594

21f1004594@ds.study.iitm.ac.in

I am from Kerala and I currently work with Uber as risk analyst

Description

The project is all about making a Grocery store application with vuejs as frontend and flask as backend.
Redis caching and celery tasks are the highlights of this project

Technologies used

The website is completely built upon python flask framework. The extensions used are flask sqlalchemy, redirect, return_template, jinja, return_template_string, request and bootstrap, flask security,redis

Sqlalchemy: For database connection
Redirect: to navigate to a particular router
Return_template: to get the html page in the router
Return_template_string: To write HTML document in return statement.
Redis for caching
Bootstrap: for styling
Flask security

DB Schema Design

Below are the important database tables:

id	product	price	quantity	category_id
Filter	Filter	Filter	Filter	Filter

1.products table, to add product and other associate details

id	category ▾1
Filter	Filter
1	2 CHOCOLATES

2.Category table to add categories

id	ısername ▾	product	category	quantity	price
Filter	Filter	Filter	Filter	Filter	Filter

3.This table is to see the users in cart products.

id	username	email	password	active	fs_uniquifier
Filter	Filter	Filter	Filter	Filter	Filter

4. To create user(part of flask security).

1. The product table and Category Management table are the core requirements for this project and the attributes given was mandatory for such a project to run smoothy and a user friendly way.

id	username	email	password	active	fs_uniquifier
Filter	Filter	Filter	Filter	Filter	Filter

2. The association table was built to make a one to many relationship between categories and products

3. The cart table was designed in such a way that the admin can clearly understand how many orders are booked for a particular category and the total revenue earned on each venues.

Architecture and Features

The backend python code is all written in main.py file. It contains database connections, flask extensions, database structure and all the @app.route definition which takes to different pages in the frontend and also some computations for dynamic pricing.

The front end code is completely rendered using vuejs. The data is transfer between frontend and backend happened with the help of fetch get or post method.
 decrease by 100rs.

Video

https://www.youtube.com/watch?v=DFfiqt0_JtI