Author

Ved Pratap
21F1000886
21f1000886@ds.study.iitm.ac.in
I am a Physics postgraduate from Patna University and aspiring Data Scientist and Programmer from IIT Madras, India. Currently gaining experience in Python, Flask, Vue.js and Backend Technologies including Machine Learning. I like to explore technologies and keep on developing new projects. I developed a keen interest in Data Science and Machine Learning so I have started studying and implementing different algorithms and analysing various datasets and started competitions on Kaggle.

Description

This project is entitled 'Grocery app' which means we have to setup a mechanism where user can explore products and can add products to cart and then they can buy it by paying some amount. Someone should be there to manage all the products. In short, there should be a role defined for every user which will give or restrict user to do some operations.

Technologies used

- Flask App
    - flask_restful – for API, flask_caching – for caching, flask_sqlalchemy – for database management, flask_security – for authorization and authentication, flask_excel – to convert some data from table to csv file
- VueJS – for User Interface (UI), Jinja2 – for templatation (not for UI), Bootstrap – for aesthetics, SQLite – for Database, Redis – for caching, Redis and celery – for batch/background jobs, Werkzeug.security – for Password hashing
- Python Module
    - datetime – for getting timestamps, dateutil – for converting string to datetime type, Numpy – for numerical purpose in creating charts for summary, matplotlib – for generating graphs/chart for summary.

DB Schema Design

1. Table: [roles users] Columns: [id, user_id, role_id] Details: ['id' is unique and 'user_id' as well as 'role_id' is foreign key]
2. Table: [user] Columns: [id, fullname, email, mobile_no, address, password, active, fs_uniquifier, roles]
3. Table: [role] Columns: [id, name, descriptions]
4. Table: [category] Columns: [id, name, descriptions]

5. Table: [manager requests] Columns: [id, man_id, type, detail, reason, is_approved, is_solved, admin_comment]
6. Table: [products] Columns: [id, man_id, timestamp, name, descriptions, cat_id, category_name, rate, unit, man_date, expire_data, in_stock, avl_quantity, sold_unit]
7. Table: [cart] Columns: [id, user_id, prod_id, name, category_name, quantity, rate, total_amount]
8. Table: [orders] Columns: [id, timestamp, user_id, orderedItems, total_amount, shipping_add, mobile_no, status, payment_status]

API Design

I have created 'resources.py' file for all the APIs. This file has end points for all the APIs which will give data after fetching from the database. I have created an endpoint 'api/products' which will return all the product data in descending order of their creation timestamp.

Architecture and Features

Architecture

Main folder contains three another folder (application, static, templates) and five files (config.py, celeryconfig.py, main.py, initial_data.py and requiremets.txt).
application folder contains all the file for backend i.e models.py has database structure, views.py has all the endpoints for all types of operations.
static has images and components for different webpages and template folder has basic template html file called 'index.html'.
config.py for configuration of app, celeryconfig.py for celery configuration, intial_data.py for creating some data when we will create database, requirements.txt has all the required packages for this project with their versions.
main.py is the file which will run our application.

Features

App users will have role as admin, manager or buyer.
Login & Signup form for Users.
Admin – Approve managers, CRUD operations with categories, Explore products, Handle requests from store managers
Store Managers – CRUD operations with products, explore approved categories, make request to admin, Download product csv file, Summary (with print button)
Buyer – Explore products, add to cart, Order, UserProfile, Check cart items & Orders
Backend/Batch jobs – Daily reminder to user who have not ordered something. Monthly report to user having all orders detail and total expenditure.
Logout option for getting out of the app.

Video

Link:
https://drive.google.com/drive/folders/16KnHhmiDw4AlZLTC74bITbyskvWbB5iD?usp=sharing