Author

Jeomon George
21f1000731
21f1000731@ds.study.iitm.ac.in

I'm a graduate of CUSAT, working as a marine engineer for Fleet Management Pvt. Ltd. I like app development. It all began in high school when I got motivated by how webpages are created, which we see on the web. This brought an interest in me to learn more about HTML, CSS, and JavaScript for web development I am not that good with frameworks but this course gave me enough information to develop this app.

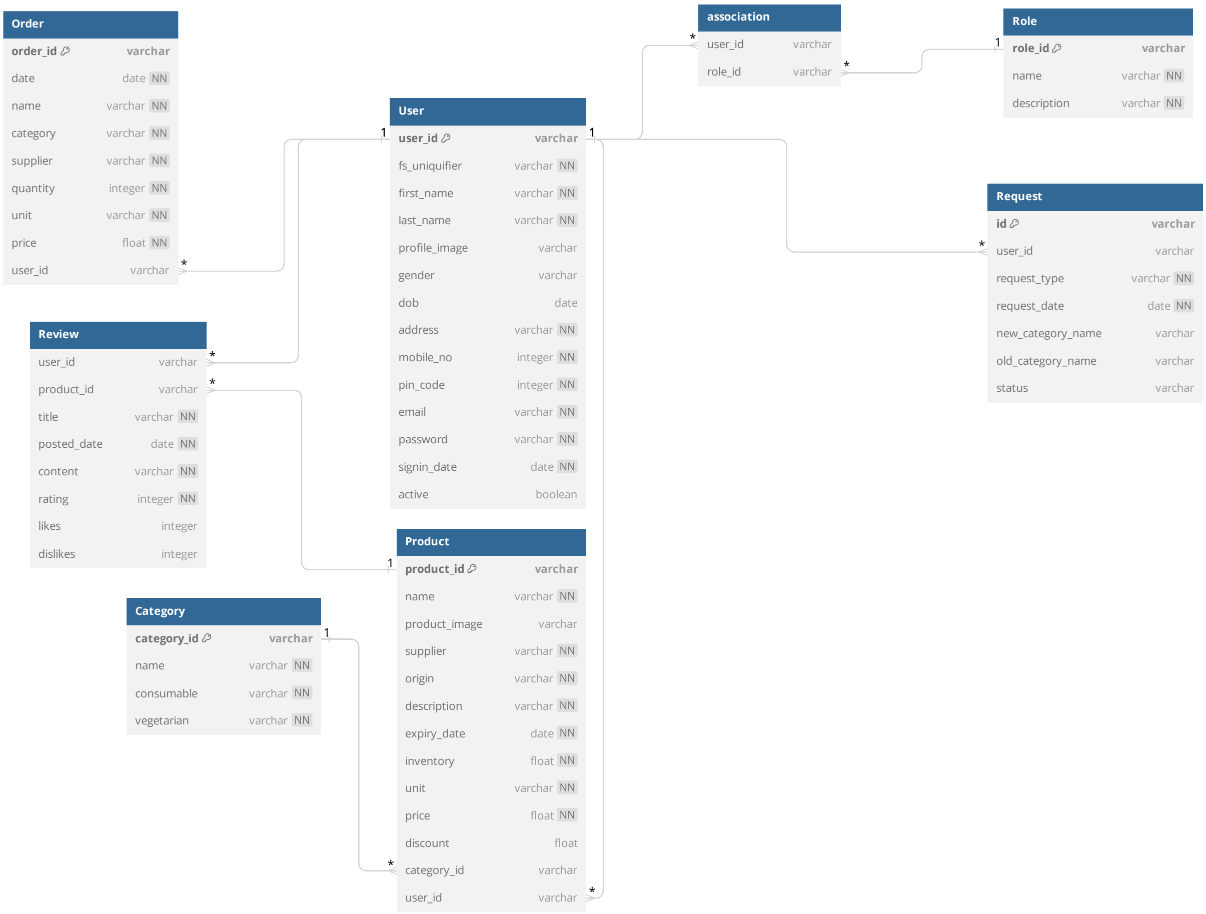
Description

The aim is to create a grocery store web app for purchasing groceries, with one admin with multiple store managers and customers. Admin adds or removes categories, and checks new store manager sign-ups and requests from store manager regarding category modifications. Customers can purchase from the store. Store managers can add new products and make requests regarding categories.

Technologies Used

- Flask: used to implement the backend.
- Flask SQL Alchemy: stored and retrieved data in the database.
- Flask Bcrypt: to encrypt the password of users.
- Flask Mail: to send emails to the users for various purposes.
- Flask Caching: to cache the data in the backend routes.
- Jinja 2: for creating templates for pdf and emails.
- Flask Security Too: for role-based access control and token-based authentication.
- Flask Blueprint: to organise each area of the app.
- Flask Cors: to enable resource sharing between servers.
- Vue CLI: implemented the frontend (with Vuex, Vue router).
- Moment: to format the date and time showing
- Axios: for making necessary API calls.
- Celery: used to handle backend jobs and scheduled periodic tasks.
- Redis: used for caching and message broker.
- Matplotlib: to show the data graphically.
- Pillow: to resize and store the image in the backend.
- os: used for accessing the directories in the file system.
- itsdangerous: used for creating tokens in email verification.
- pdfkit: used to create pdf for reports

DB Schema Design



Order		
order_id	**varchar**	
date	date	NN
name	varchar	NN
category	varchar	NN
supplier	varchar	NN
quantity	integer	NN
unit	varchar	NN
price	float	NN
user_id	varchar	

association	
user_id	varchar
role_id	varchar

Role		
role_id	**varchar**	
name	varchar	NN
description	varchar	NN

User		
user_id	**varchar**	
fs_uniquifier	varchar	NN
first_name	varchar	NN
last_name	varchar	NN
profile_image	varchar	
gender	varchar	
dob	date	
address	varchar	NN
mobile_no	integer	NN
pin_code	integer	NN
email	varchar	NN
password	varchar	NN
signin_date	date	NN
active	boolean	

Request		
id	**varchar**	
user_id	varchar	
request_type	varchar	NN
request_date	date	NN
new_category_name	varchar	
old_category_name	varchar	
status	varchar	

Review		
user_id	varchar	
product_id	varchar	
title	varchar	NN
posted_date	date	NN
content	varchar	NN
rating	integer	NN
likes	integer	
dislikes	integer	

Category		
category_id	**varchar**	
name	varchar	NN
consumable	varchar	NN
vegetarian	varchar	NN

Product		
product_id	**varchar**	
name	varchar	NN
product_image	varchar	
supplier	varchar	NN
origin	varchar	NN
description	varchar	NN
expiry_date	date	NN
inventory	float	NN
unit	varchar	NN
price	float	NN
discount	float	
category_id	varchar	
user_id	varchar	

API Design

User API: to sign up, sign up, sign out, update and delete user.

Category API: to create, delete, update, and access categories.

Product API: to create, delete, update, and access products.

Request API: to create, delete, update, and access requests.

Cart API: to add, delete, update, and access products in the cart.

Order API: to add, and access recent purchases.

Review API: to add, delete, and access the reviews for a specific product.

Architecture and Features

All the routes, models, additional files...etc. for an API in the backend, it is kept in a folder with its name as an API. The Jinja templates are only used for creating emails and are all kept in the templates folder of the application module. Graphs will be generated and kept in the static folder of the application module and the generated CSVs will be temporarily stored in the temp folder in the root directory.

In the frontend all routes are grouped based on API names in the router module and the state is maintained in the store module of Vue CLI. Views for routes are kept in the respective API-named folders. Some common elements used in the views are kept in the components folder.

- Login System: here, the user can create a new account based on role and log in
- Category Management System: here, the user can do CRUD to categories
- Product Management System: here user can do CRUD to products
- Search Engine: here users can get products based on categories and product name
- Export Jobs: export CSV and graphs for products based on the category to the store manager
- Report Jobs: generate monthly reports and send them to customers via mail
- Alert Jobs: send emails to those customers who are not so actively visiting
- PDF Report: the monthly email reports have PDF reports as well
- Additional features: review and rating with its statistics for each product, customers can see the recent orders.

A Tour Through the App

Video link: Click Here