

Author

Divyansh Raj

21f1006675

21f1006675@ds.study.iitm.ac.in

I am currently a diploma student in this IIT Madras BS degree with a strong interest in data science and machine learning. I am looking forward to pursue my career in data science and machine learning field.

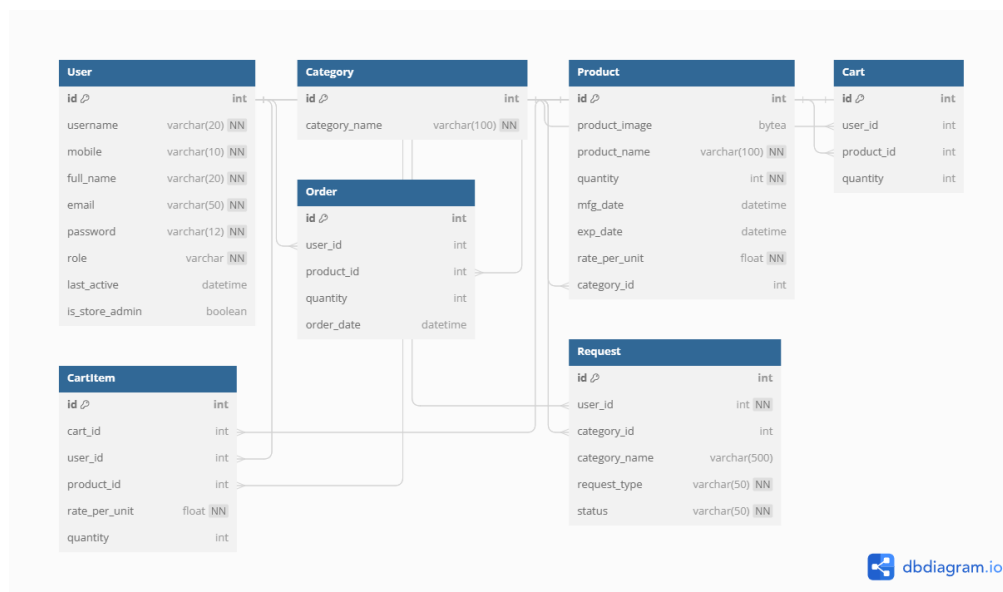
Description

This grocery shopping application is designed to accommodate three user roles, each with specific functionalities. The primary admin has full control over categories, allowing them to create, read, update, and delete categories. Store admins, on the other hand, can manage products within the categories set by the admin, along with the ability to request changes to existing categories. Users can request to become store admins. This tiered system ensures efficient management and a seamless shopping experience for all users. Users can get daily reminders and monthly report.

Technologies Used

Technology	Purpose
Flask	Micro framework
Flask SQLAlchemy + SQLite	Data Storage
JWT	Route protection and role verification
Vue3 + JavaScript	Frontend
Flask Restful	Api endpoints
Flask CORS	Connecting Backend to Frontend
Bootstrap	CSS Components
Celery	To schedule periodic asynchronous tasks
Redis	Inmemory database for celery and caching
Flask Mail	Mailing Reminders
	Etc.

DB Schema Design



Api Design

Api has been made using flask in restful manner.

Architecture & Features

The backend has been made using Flask framework in a virtual environment “**venv**”. Api requests like get, post, put and delete has been written in restful manner. Route protection has been done using Json web token (JWT). Api for every endpoint is in a separate file in “**src**” folder. Frontend has been made using vue3 with CSS library Bootstrap for components and responsiveness and Font Awesome for icons. For frontend some reusable components are in components folder and views are views folder.

Command to run backend server : `python main.py` (flask app)
`redis-server` (redis server)
`celery -A src.celery_workers.celery worker --loglevel=info` (Celery worker)
`celery -A src.celery_workers.celery beat --loglevel=info` (Celery beat)

Command to run frontend : `npm run dev` (Development server)

Features Implemented

USER

User can buy, search for products, filter products according to categories, see order history, get daily activity mail and monthly report mail.

ADMIN

Admin can perform CRUD on categories and view products in inventory and approve/reject requests from users to be store admin and requests from store admin to manage categories.

STORE ADMIN

Store admin can perform CRUD on products and request admin to manage categories.

Presentation Video

[Link](#)