# Grocery Store Application v2

# Modern Application Development 2

## Author:

Sakiley Pranay Deep

Roll number: 21f1005603

Email: 21f1005603@ds.study.iitm.ac.in

## App Description:

This application has an admin, managers and customers. Managers can create/edit/remove categories and products, while customers can search and shop for the products (adding to cart or buying the products). Admin has to approve the sign up requests and category requests from the managers. Application sends automated reminders and reports to users based on their activity.

## Technologies used for the project:

The backend of this application works mainly on Flask framework for python, while the frontend is made using Vue.js framework. Celery and Redis are used for backend jobs and caching. SQLite3 is used for database.

More technologies used in this project include Flask-RESTful, Flask-SQLAlchemy, Flask-JWT-Extended, werkzeug, Flask-CORS, Flask-Mail, Pillow, Flask-Caching etc.

## Database Schema Design:

User model: Stores the id, username, password, role, approval status (valid for manager sign ups) and the time of last purchase for each user.

Manreq model: Stores the usernames of the managers that are yet to approved by the admin.

Category model: Stores the id, name and description for each category.

Catreq model: Stores the id, name and description of each category request from the managers, along with the request type (create/edit/delete) and the user id of the manager that made the request, as a foreign key.

Product model: Stores each product’s id, name, unit, cost per unit, image name, manufactured date, usage days, quantity in stock and the category to which it belongs as a foreign key.

Cart model: Stores the id of each addition to the cart, along with the ids of the corresponding product and user as foreign keys, and the quantity of the product added.

Order model: Stores the order’s id, id of the user that made the purchase as a foreign key, name of the product purchased, quantity of the product bought, total cost of the order and the timestamp of the order.

## Architecture and features:

This project uses Vue.js and its templates for the front-end, which make requests to the back-end flask application to fetch the data as needed. In this application, users are categorised into three roles: admin (one admin added manually when the database is created), managers and customers. Users can sign up as customers or managers. Admin is above the managers. Admin can create/edit/delete categories and products, and supervises managers’ sign up requests and category create/edit/delete requests. Managers can create/edit/delete products without needing approval. All the users get newly added products directly onto their dashboards. All users can view products/categories by visiting the corresponding pages. While admin/managers are able to edit/delete existing products, customers are able to buy them or add them to cart. The cart page allows customers to buy all the products in the cart at once. There is a dynamic search functionality, where users can search for products based on name, category and price. Celery and Redis are used to implement scheduled jobs, which send a reminder mail to all the users who haven’t made any purchase in the last 24 hours. A monthly report of the shopping history to the customers who made purchases is scheduled, too. Managers and the admin are able to export the product details as csv through the dashboard. Caching is implemented at several parts of the application to improve its performance.

## Video presentation link:

https://drive.google.com/file/d/15jh2QUGF-9DnCfg7euIEbHyLKud6VWCv/view?usp=drive_link

## Thank you!