

# MAD-II Project

Name : Jay Thadeshwar

Student Mail Id : 22dp3000018@ds.study.iitm.ac.in

Roll Number: 22dp3000018

## Title: Grocery Store App

1. **Introduction** : Grocery Store App is a web based application to buy Grocery online. Title of my App is: Nature's Nourish.
2. **Overview**: I have undertaken this project with the goal of developing a web-based inventory management system for the grocery store. Primary objective of this project is to create a user-friendly platform that enable people to buy product from grocery store online.  
My project focus on developing web application using flask by which user can buy products of grocery store online and admin can manage products and category on web application
3. **Technologies Used**:  
Flask framework: used for building application's backend.  
Flask-Security: For proper Login system.  
Bcrypt: For hashing password.  
SQLite database: For data storage and retrieving  
Bootstrap: For styling  
Jinja 2: For creating Monthly report  
Flask-SQLAlchemy: For Database Integration  
Flask RESTful: To design and implement Restful API endpoints.  
Vue Js.(CLI) : Used for Building application's front.  
Redis: For Caching.  
Redis and Celery: For batch jobs.  
MailHog: For sending Emails.  
Weasy print: For Making pdf
4. **Data Models**:
  - **User**: for storing user's information.
  - **Roles**: for storing role's information.
  - **Category**: stores category's information like name description..etc..
  - **Product**: stores products's information like name price..etc.

- **Cart:** stores users cart's information like how many product they have added into cart and it's quantity.
- **Order:** stores user's orders info like total, address, name ..etc
- **Order\_Details:** stores details of orders like how many products and its quantity, price
- **CreateReq:** stores manager's request to admin to create category
- **DeleteReq:** stores manager's request to admin to delete category
- **EditReq:** stores manager's request to admin to edit category
- **LoginReq:** stores manager's signup request.

5. **System Architecture:** the system follows client-server architecture. the client side is built using vue.js and server-side is built using flask to handle api requests and interaction with database.

#### 6. **Functionality :**

- I have implemented a role-based access system. And app have three roles: Admin, Store Manager and User.
- Users can visit products, search products, visit products by category, add products into the cart and buy the product.
- 
- Manager can perform CRUD operation on products, can see all the orders made by the user, Manager can delete orders, request admin for create, update and delete categories. And the manager can download the list of products in csv format. I have implemented this as an Asynchronous task.
- 
- Admin can approve a manager's request for signup as a manager, Admin can create , update, delete categories with or without manager's request.
- App can send daily reminders to the user via email.
- App can send Monthly report to the users in html format via email.
- My app is using redis for caching to improve the performance of the app
- I have added styling to make my app look good.
- Yaml file for api documentation for CRUD operation on category.

#### 7. **Video Link:**

[https://drive.google.com/file/d/1yO4NkAs3\\_NdVlcMf90DvvtH2NMyjQ0BJ/view](https://drive.google.com/file/d/1yO4NkAs3_NdVlcMf90DvvtH2NMyjQ0BJ/view)