**Description**

The Project majorly focuses on building Grocery Store App, where we provide an interface such that it caters to the people managing the store online, that is allowing them to handle various products and categories, as well as the people who are buying products from this app, giving them an opportunity to explore what is offered.

**Technologies used**

Flask Extensions : Flask, render\_template, request, redirect, SQLAlchemy for creating functions and maintaining the core of an app such as switching between different web pages and merging HTML with python and flask.

Datetime Extension for using the date and time template for the various date types on the website.

JavaScript, HTML and CSS for styling and implementing web pages

SQLite for maintaining a database, such that it is easier to work on multiple databases such as Product, Category, Admin and Users.

**DB Schema Design**

1. **Category :** The Category DB essentially provides a unique ID to each category and gives a name to it. The products relation is used to give each category it’s separate list of products.
2. **Product :** The Product DB covers the ID, name, price and manufacturing and expiry dates of each product. It also helps to connect each product with a category.
3. **Admin :** The Admin DB lets us track and store the data for all the managers logging into the Grocery App to make changes to various categories and products.
4. **User :** The User DB lets us track and store the data for all the buyers/potential buyers logging into the Grocery App, allowing us to maintain a database to keep them engaged and for future references.

**Architecture and Features**

The project is contained as a zip file, with 3 folders and a Python Source file. The python file, called app, is the main python file with various functions for each functionality as well as the model for each database created. The first folder, instance contains the SQLite Database containing all records for all 4 models. The second folder, static contains the JavaScript codes, specifically for cart. The third folder, template has about 18 HTML codes, pertaining to each web page and service offered.

The core features are implemented in the project, where there are separate login pages for the Admin and User. The Admin can manage all categories and products, that is, create, edit, view and remove any and all of them. For users, as they login, they are taken to the store which displays a search bar and the list of all products offered.

These functionalities are implemented using basic HTML coding, use of flask extensions and python functions. Minimal to no CSS is used, for implementing tables and minor aesthetics to the main page for users.