**Capstone Project 1: Local food business finder app**

The website that I am designing and planning to program is intended to recommend a user a list of good local food service businesses. It also shows a list of non-recommended businesses as well. When it comes to the user demographics, this site aims at users of all levels and all backgrounds. For this site to best serve a user’s need, an external API will be used. This API contains data about local businesses. My work would be to fetch data about local business from that API and filter to generate data related to local food service providers including restaurants and fast-food marts. In addition, based on the reviews/ratings, I am intended to generate a list of recommended stores as well as a list of non-recommended ones that a user can choose from. User can search a store by typing in a city name. If an API provides a benefit of searching by zip code, then the site will incorporate that one too.

My approach on creating this site includes having a home(landing) page. This landing page just contains a brief description of what this site is about, and then navigation bars to link to different pages like signup page, login page, user favorite page, detail page, and profile page. Profile page renders an html form where a user can update her information. Login page should show list of different favorite stores a user has after a user logs in. When the user clicks on a particular store name, it should take that user to the page where it renders details about that store. This functionality should do the exact same thing even when a user clicks on the store name on her favorite list.

Since this is a data driven project, I must have a way to store certain user data. My plan includes creating a database to store user meta data such as a username and hashed password so that they can be used for authentication and authorization purposes. In addition, I also need database to store user’s favorite stores and details about those favorite stores. At this point, I am assuming that a details about a store that a user has in a favorite list is dynamic as it comes from an external API. All I will do is try to render some of the useful information about that store such as average price per dish, delivery options, address, phone number, and hours of operation. The database schema includes three relational tables, one called users, second one called stores, and third one named favorite stores. They represent many to many relationships as follows:

A user can choose many stores. A store can have many users.

While in the process of creating this site and deploying it, I may run into issues such as API going down momentarily for things like data update on their site, an API may simply go offline, and it may come with a restriction of a usage rate limit such as certain times a day. Other issues may include invalid API key error as well as validation error. I need to be careful to store a user’s password into a database. My intention is to use Flask-BCrypt algorithm to hash it before sending a password to store in a database so that it provides password security. Another sensitive information would be my API key, that I decided to implement through only back-end.

When a user goes to a website URL, it shows a home page. The home page contains links to different pages such as signup page, favorite page, profile page, and login page. When a user signs up, it will render a search bar, in which a user can type a city name to search food stores. When a user likes a store, he or she can favorite that store. For the first time user, it should create a favorite list for that user, and add that store to the favorite list. For the repeating user, it simply should add a store to the favorite list. Clicking on a favorite store should show details about that store.

The idea of fetching data about local businesses deals with CRUD concept. The process of filtering the data to generate list of local food service providers goes beyond the CRUD implementation. Also, filtering and rendering only most useful information about a store goes beyond CRUD work. Similarly, computation related to recommending list of stores based on certain rating criteria also goes surpasses the CRUD implementation.

I NEED HELP ON THE FOLLOWING PLEASE:

users