August 3, 2023

CP1895

Python Project Plan

Kyle Saunders

20052402

**Part 1: Web Application Overview**

(Any updates since original submission are highlighted with the \* Update \* symbol in brackets where appropriate.)

**A. Pages Structure and Content**

The proposed web application will consist of five primary pages:

1. **Homepage**: An engaging welcome page with an introduction, key features, navigation links, and a section to highlight featured or popular recipes.

2. **Recipe Gallery**: A page to showcase various recipes with pictures, ingredients, preparation steps, serving guidelines, and average ratings.

(\* Update \* I created another page called recipe\_detail.html early in development thinking I would use it to re-direct the user to a html page where it displayed the recipe details, but as I researched in JavaScript I discovered modals and used a modal to pop up and display recipe cards rather than dedicate a page to it which would result in more connections to the server that the user would need to navigate through. The modal is much more efficient and aesthetically pleasing in my view. I deleted this file in the final commit.)

3. **Recipe Submission**: A user-friendly interface for submitting new recipes, including image upload, ingredients, preparation, and serving instructions.

4. **Recipe Management**: A specialized page to enable users to delete recipes.

5. **Search Functionality**: A dedicated search page with a search bar and an ingredient filter, allowing users to search for recipes by name or specific ingredients.

**B. Template and Dynamic Content**

Jinja2 templates will ensure a consistent look across all pages, while dynamic content such as recipe details and user ratings will be fetched from CSV files.

(\*\* Update \*\* I also used a couple of JavaScript functions to handle the dynamics of the modal “Recipe Card” that pops up when you click the View Recipe button on any of the HTML pages.)

**C. Navigation Structure**

A navigation bar will be present at the top of each page, containing links to all pages. The current page will be highlighted for easy reference.

(\*\* Update \*\* I decided to just go with titles atop each page to let the user know which page is current.)

**D. Design and Deployment Goals**

The goal is to achieve a high-quality design (90-100% on the rubric), utilizing Bootstrap for styling, ensuring mobile-friendly design, and deploying on platforms like Python Anywhere or AWS. The secondary target is 80-90%.

**Part 2: Technical Specifications**

**A. CSV File Format**

The CSV file will include columns for recipe name, image link, ingredients, preparation instructions, serving instructions, and user ratings.

**B. Optional Technologies**

Bootstrap will be employed to enhance the site's responsiveness, user experience, and mobile accessibility.

**C. Deployment Strategy**

The web application will be hosted on Python Anywhere.

**D. Static Files Inventory**

Static files to be used include:

- HTML files for each page

- CSS file for styling

- Recipe images

(\*\* Update \*\* Added a JS file here as well)

**E. User Interaction Features**

- User Profiles and Ratings: Allow users to create profiles and rate recipes.

(\*\* Update \*\* I ran into some debugging issues when attempting to implement user profiles so I decided against including this feature. I did come up with a way for anyone to rate the recipes who visits the site as a guest.)

- Comment Section: Add a comment section to each recipe page for user interaction.

(\*\*Update\*\* I did not include this functionality in the final version either due to the lack of user profile.)

- Ingredient Filter: Implement a filter to find recipes based on ingredients.

(\*\* Update \*\* Users can search by name or ingredient.)

(\*\* Update \*\* I also added the ability for a user to upload their own recipes and images to go along with those recipes.)

**F. Additional Resources**

- Stack Overflow Questions about PythonAnywhere: https://stackoverflow.com/questions/tagged/pythonanywhere

- How to make a web Application using Flask:

https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3

- Bootstrap 5 Tutorial: https://www.w3schools.com/bootstrap5/