Search Criteria:

The user will be able to search for recipes by searching the recipe name or ingredients in the recipe.

Data Visualization:

Bar Chart- Displays recipes alongside each other by cooking time.

Pie Chart- Displays recipes by difficulty.

Line Chart-Display the number of recipes created per day

Execution Flow:

- User lands on home page
- Clicks login and enters login information
- User lands on Recipe Overview list
- User searches for ingredient
- User views various charts (data analytics) for the search
- User returns to the list of all recipes
- User clicks on a Recipe to view details
- User returns to list of all recipes
- User logs out

Reflection Questions:

1. Consider your favorite website/application (you can also take CareerFoundry). Think about the various data that your favorite website/application collects. Write down how analyzing the collected data could help the website/application.

If you think about the data that Netflix collects, such as what shows the user has viewed recently, that data would be very helpful in showing other shows that the user might be interested in. The more accurate shows Netflix can refer to the user the more time the user will spend viewing their application.

2. Read the Django official documentation on QuerySet API. Note down the different ways in which you can evaluate a QuerySet.

Iteration-Loop through a QuerySet to access each object individually.

Slicing- Slicing an unevaluated QuerySet usually returns another unevaluated QuerySet, but using the ‘step’ parameter will return a list

Pickling/Caching-Involves reading results from the database.

repr()-this causes the QuerySet to be evaluated.

len()-Retrieve the length of the list.

list()-Converting a QuerySet to a list forces its evaluation.

bool()-Evaluates a QuerySet in a boolean context and executes the query.

3. In the Exercise, you converted your QuerySet to DataFrame. Now do some research on the advantages and disadvantages of QuerySet and DataFrame, and explain the ways in which DataFrame is better for data processing.

-A DataFrame offers more flexibility and functionality than QuerySets. They also offer faster memory processing and better integration with Python Libraries. However DataFrames consume more memory and may not always reflect real-time data changes, QuerySets are more suited for large data sets.