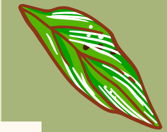
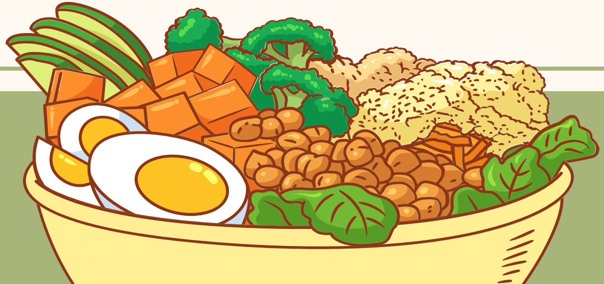


Recipe Generator

Aviva Munshi
Nizan Howard
Saumya Nauni



Background

The problem being addressed in our project is the difficulty in deciding what to cook with the ingredients you have at home.

The goal is to build a streamlined web app that can generate a recipe from a collection of recipes.

The input required is a list of ingredients that the user has at home.

The output will be a random recipe that uses the ingredients listed.

Data Used

We used two websites, **epicurious** and **bonappetit**.

Web scraping was employed using **BeautifulSoup** and **recipe-scrapers.**

Number of recipes scraped: Approximately **400.**

Limitations of the data:

- Bonappetit's search page has a continuous scroll which made getting recipes difficult.
- Various ads and food articles included in the websites were parsed as recipes, causing initial scraping trouble.

Use Cases

Use Case 1:

- **Objective:** The user receives a recipe that includes the ingredients they provide upon input.
- The user types in the ingredients they have, and the generator checks if they are valid for human consumption.
- If valid, the generator displays a recipe that includes the input ingredients.
- If not valid, the generator displays an error message and reports that it couldn't find a recipe for the listed ingredients.
- The recipe consists of instructions on how to make the dish using the input ingredients.
- The user uses the instructions to prepare the dish.

Use Cases

Use Case 2:

- **Objective:** The user receives a recipe that matches their specific dietary requirement.
- The user selects their dietary requirement, enters the ingredients they have on hand, and the desired meal time.
- The application searches for a recipe that matches the user's dietary requirement, the provided ingredients, and the desired meal time.
- The application displays the recipe that matches the user's dietary requirement, uses the provided ingredients, and matches the desired meal time.
- The application displays relevant information such as cooking time, calories, and serving size.
- The user reviews the recipe and relevant information.

Use Cases

The system addresses the problem area of recipe selection and meal planning by providing users with personalized recipe recommendations based on their specific needs and preferences.

Use Case 1 allows users to generate a recipe for a specific amount of ingredients, which can help reduce food waste and save money by using what they already have.

Use Case 2 enables users to generate recipes that match their dietary requirements and desired meal time, making meal planning easier and more efficient.

Overall, the system streamlines the process of recipe selection and meal planning, which can help users make healthier and more cost-effective food choices.

Design

The system consists of three components -

A recipe database

A web page to collect user input, and

An **input-output mapper** that maps user input to the database to retrieve the corresponding webpage output.

Design

Component 1

The **Recipe Database** component collects recipes from web pages and stores them in a dictionary.

The input is a subset of web pages that have recipes, and the output is a list of dictionaries that includes recipes and links to the original web page.

Design

Component 2

The Web App component provides an interface to collect inputs from the user, such as dietary requirements and ingredients.

The output is a recipe that uses the ingredients mentioned in the input. The assumptions are that ingredients should be food items with basic condiments as a staple in everyone's pantry and that there should be at least 3-4 ingredients listed to get an appropriate recipe.

Design

Component 3

The **Input-Output Mapper** component maps user inputs from the Web App to the corresponding database entries and retrieves the corresponding webpage output. The inputs are user input from the Web App and a database with recipe information (e.g., recipe ingredients, preparation instructions).

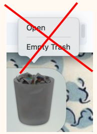
The output is a webpage output containing the recipe that meets the user's input criteria (e.g., recipe name, ingredients, preparation instructions).

Demo

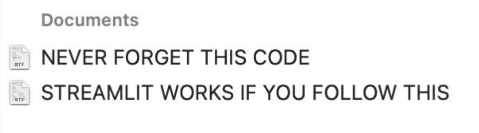
https://drive.google.com/drive/folders/1ZlpgV7tc2d-A-urfXKiRuAKE2tt-RXYA?usp=sharing

Lessons learned and future work

- Don’t put off Continuous Integration till the end!!
- Don’t ever empty your trash



- Good documentation and comments are important for making code understandable and maintainable. In our case, maybe don’t name files like this:

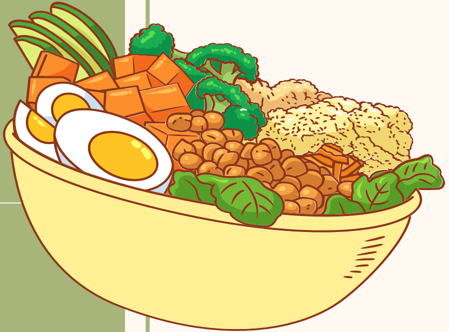


Documents

NEVER FORGET THIS CODE

STREAMLIT WORKS IF YOU FOLLOW THIS

- Improving the user interface and user experience of the web application to make it more intuitive and user-friendly is important.
- Using machine learning algorithms to improve the accuracy and relevance of recipe recommendations.



Thanks!

Do you have any questions?