Django Recipe
APP

Start Cooking!

- Login Here -

Project Overview:

Role: Lead Developer
Mentor: Azamat Uzebekov

Tools, Skills, & Methodologies:
- Python 3.6+
- Django (version 3)
- PostgreSQL database
- SQL database

My recipe app is a web application with multiple users and an admin panel, using Python-based Django framework. The goal of the app to allow users to create and modify recipes. Essentially, an online recipe book for the modern cook or baker that prefers to not keep cookbooks.

Key Features:
- Difficulty parameter automatically calculated by the application for users that choose recipes based on their difficulty
- Ingredient-based search option for users that only have specific ingredients

Purpose and Context
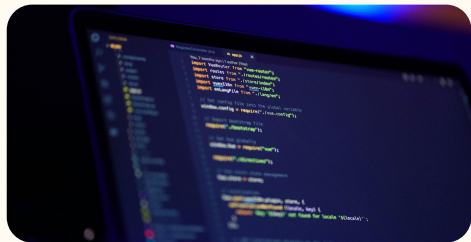
This was a personal project I built as part of my web development course at CareerFoundry to demonstrate my understanding of Python, Django, and HTML.

Objective

- To create a full-stack project that I could add to my professional portfolio
- Problem to be solved:
  - to build the complete server-side and client-side for the application, while utilizing tools like a Postgres database and Heroku.

Duration

Compared to other JavaScript projects I've completed; it took me considerably less time to complete this project. I think a large part of that has to do with the similarities between Django and JavaScript

The approach started with learning the basics of Django, like the importance of file structure and its implementation with Python, and when it should or shouldn't be used. With this information I created the basic Django project structure and a superuser to perform administrative tasks, like adding development data and/or users.
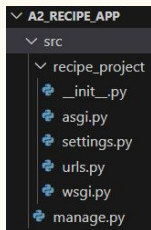
I was given a project brief with a set of technical requirements and application features. Using the project brief and other recipe applications as a reference, I used Django's MVT (Model View Template) architecture to create the home page, various subpages, and login/logout pages.
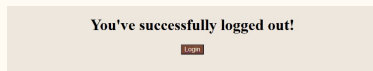
One of the subpages I created was a search page where the user could search for recipes by an ingredient name. In addition to the user seeing the results from their search, the type of chart of their choosing is also displayed on the page.

As I created each view and template, I wrote tests to ensure that functionality of each page was as I expected it to be. Finally, before I deployed my application using Heroku, I added a navigation bar that includes a link to my professional portfolio.

STEPS

FINAL THOUGHTS

Takeaways & App Improvements

One of the changes I made to my application to deploy it was remove the images from that recipes as my tutor informed me that Heroku plan I was using didn't support saving static files. In the future, I plan on updating my Heroku plan and add the images back to the recipes because I liked the look of the application that way better. I look forward to continuing to work with Django on this project and others in the future.

What didn't go well?

The most difficult challenge I experienced with this project was deploying it to Heroku. The database the application was working with in deployment was SQLite, which is a database that Heroku doesn't work with. In order to deploy my application using Heroku, I needed to update my application to use a Postgres database post-deployment. While I made these changes locally on my application, I needed to also add the add-on Heroku Postgres to my application via Heroku.

What went well?

Django's use of MVT architecture and Django's similarity to JavaScript made it so I was able to understand Django much quicker than I was able to understand JavaScript. My quicker understanding of Django than JavaScript and its strict structure allowed me to enjoy creating the application more. Additionally, I think this project rounds out my professional portfolio as most of my projects are JavaScript based.