

Recipe App Case Study

A Django & Python Project

Patrick Kernan

pkernan18@yahoo.com

Visit my Portfolio Site for More!

Recipes

[Contribute a Recipe](#)

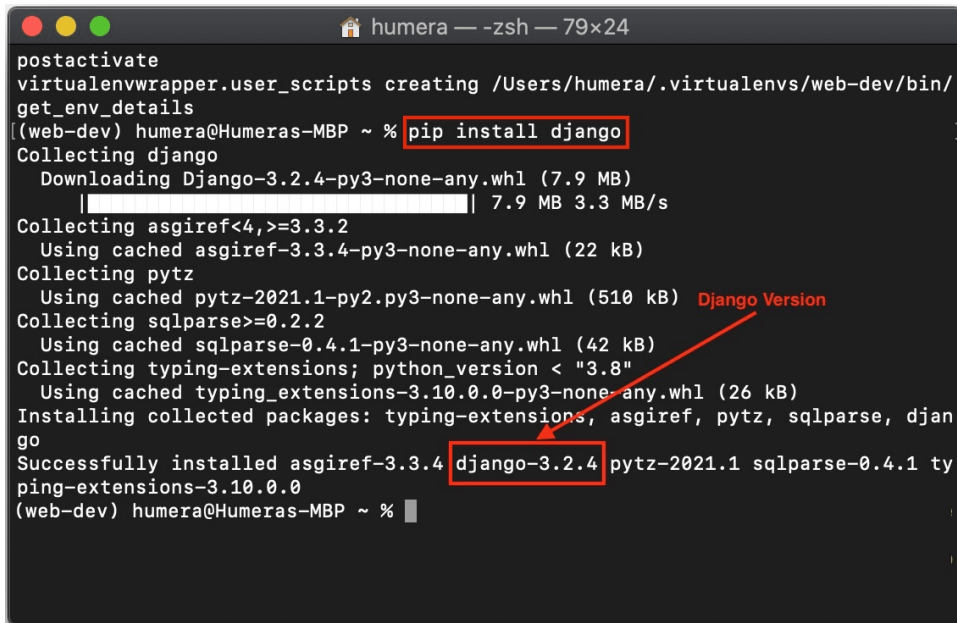
[Logout](#)

Name	Ingredients	Cooking Time	Difficulty	Instructions
Patrick's Struggle Raman	Shin noodles, peanut butter, gochujang, garlic powder, onion powder, cheese	4 min	Medium	View Instructions
Mac & Cheese	macaroni noodles, milk, cheese	15 min	Intermediate	View Instructions
Spaghetti and Meatballs	spaghetti noodles, ground pork, tomato sauce	15 min	Intermediate	View Instructions
Pepperoni Pizza	pizza dough, pepperoni, cheese, tomato sauce, olive oil	30 min	Hard	View Instructions
Tea	tea leaves, water	2 min	Easy	View Instructions
Grilled Cheese	bread, cheese, butter	15 min	Intermediate	View Instructions

Challenge, Action, Result (CAR)

Challenge

I set out to build a simple yet functional recipe application where users could create an account, log in, and access a database of food recipes. The challenge was to apply my newly acquired skills in Python and Django to create a project that wasn't just a tutorial exercise, but something I could design, build, and expand upon myself.



```
humera — zsh — 79x24
postactivate
virtualenvwrapper.user_scripts creating /Users/humera/.virtualenvs/web-dev/bin/
get_env_details
[(web-dev) humera@Humeras-MBP ~ % pip install django]
Collecting django
  Downloading Django-3.2.4-py3-none-any.whl (7.9 MB)
    | 7.9 MB 3.3 MB/s
Collecting asgiref<4,>=3.3.2
  Using cached asgiref-3.3.4-py3-none-any.whl (22 kB)
Collecting pytz
  Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.1-py3-none-any.whl (42 kB)
Collecting typing-extensions; python_version < "3.8"
  Using cached typing_extensions-3.10.0.0-py3-none-any.whl (26 kB)
Installing collected packages: typing-extensions, asgiref, pytz, sqlparse, djan
go
Successfully installed asgiref-3.3.4 django-3.2.4 pytz-2021.1 sqlparse-0.4.1 ty
ping-extensions-3.10.0.0
(web-dev) humera@Humeras-MBP ~ %
```

A red arrow points from the text "Django Version" to the "django-3.2.4" package name in the installation output.

Action

To meet this challenge, I learned the fundamentals of Django, including app structure, template rendering, URL routing, and form handling. This included everything from the simple step of installing Django and learning how to use a virtual environment (something I was completely unfamiliar with.) CareerFoundry taught me the basics of creating a Django project, creating apps within the project, and what these terms really mean. It helped to view Django as a straightforward sort of building project. The apps are just pieces of the whole, which I could easily 'plug-in' to the main project. CareerFoundry offered a great deal of freedom to experiment, which helped me to not only build my base knowledge, but also learn how to use resources (documentation, AI, etc.) to expedite my workflow. I then applied this knowledge to incrementally build my own recipe app. Starting with a basic home page and recipe list, I expanded the app with login functionality, user registration, and recipe creation forms.



Result

The end result was a fully functioning recipe website where registered users can create an account, log in, and upload their own recipes to share with others. While the design is intentionally simple, the project serves as a strong foundation for my growth as a developer and demonstrates my ability to take an idea from concept to implementation. Some of my recent attempts to build on this have been learning how to use and integrate CSS to give things like my recipe form a more user-friendly feeling. I've begun to view this exercise as a good exercise in true Full-Stack development, working on both the user experience and the inner workings of the app itself.

```
recipe_project > views.py > ...
1  from django.shortcuts import render, redirect
2  from django.contrib.auth import authenticate, login, logout
3  from django.contrib.auth.forms import AuthenticationForm, UserCreationForm
4
5  def login_view(request):
6      error_message = None
7      form = AuthenticationForm()
8
9      if request.method == 'POST':
10         form = AuthenticationForm(data=request.POST)
11         if form.is_valid():
12             username = form.cleaned_data.get('username')
13             password = form.cleaned_data.get('password')
14             user = authenticate(username=username, password=password)
15             if user is not None:
16                 login(request, user)
17                 return redirect('recipes:recipe-list')
18             else:
19                 error_message = 'oops.. something went wrong'
20
21         context = {
22             'form': form,
23             'error_message': error_message,
24         }
25         return render(request, 'auth/login.html', context)
26
27  def logout_view(request):
28      logout(request)
29      return redirect('login')
30
31  def register_view(request):
32      form = UserCreationForm(request.POST or None)
33      if request.method == 'POST' and form.is_valid():
34          user = form.save()
35          login(request, user)
36          return redirect('recipes:recipe-list')
37
38      context = {
39          'form': form,
40      }
41      return render(request, 'auth/register.html', context)
42
```

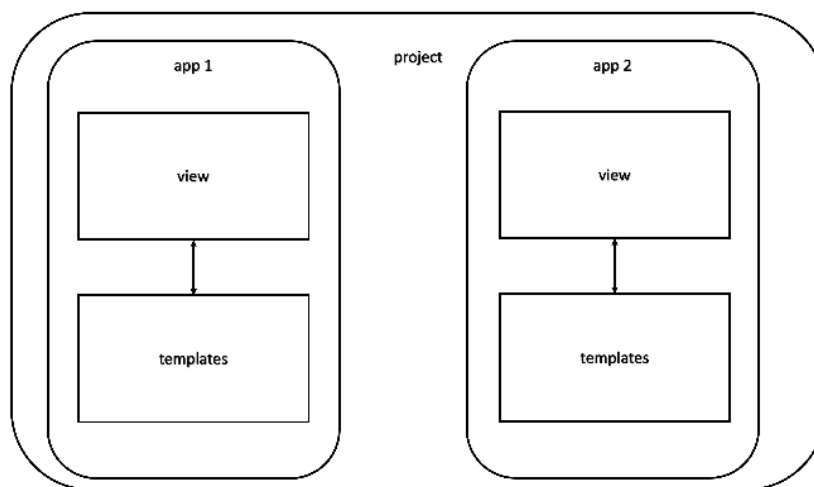
Process Breakdown

Step 1: Learning the Basics

I began by learning the Django framework — setting up apps within a project, working with templates, and connecting URLs. This gave me the scaffolding I needed to start building a real project. Simple, given steps like when to create an app and when not, where all new concepts to me. I was able to recognize the patterns in Django's logical framework, which made things much easier.

Step 2: Building the Foundation

I created a home page that displayed a list of recipes managed through Django's admin interface. This helped me understand how models, views, and templates connect.



Step 3: Implementing Authentication

To allow users to access recipes, I learned how to use Django forms to create a login page. This was my first exposure to integrating user authentication into a project. Altering the settings in the specified Python file gave me insight into the potential complexity of apps built in a framework like Django.

Login

Username:

Password:

Don't have an account? [Register here.](#)

Step 4: Expanding Features

With the basics working, I began planning additional features to make the app more interactive. The most impactful addition was creating a registration page so that new users could easily sign up with just a username and password. I have always appreciated websites with minimal flipping between different pages and simple registration processes. This also allowed me to, for the first time, send my project out to friends and family who weren't knowledgeable about coding to test it for themselves and provide me feedback on their experience trying it.

Step 5: Enabling User-Generated Content

To make the app more dynamic, I added a recipe form where registered users could create and upload their own recipes. This turned the app from a static recipe list into a collaborative platform.

Contribute a Recipe

Recipe Name:

Ingredients (separated by commas):

Cooking Time (in minutes):

Instructions (optional):

Recipe Image (optional): No file chosen

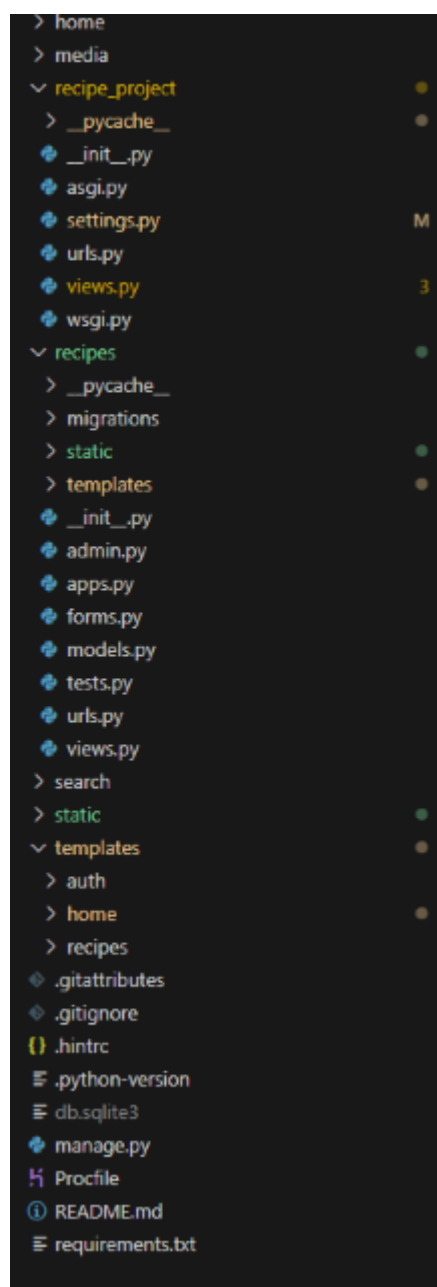
[Back to recipes](#)

Retrospective

Building this recipe app was both a technical and personal milestone. It gave me practical experience with Django's ecosystem, from handling templates and URLs to implementing authentication and user-generated content.

The project also taught me how to approach development iteratively — starting with a simple foundation and gradually layering on functionality. Even though the current design is straightforward, I now see it as a launchpad for more ambitious projects.

Most importantly, the process gave me confidence. Learning how to solve problems, test my work, and build something functional from scratch left me with a strong sense of accomplishment. Going forward, I plan to iterate further by adding search functionality, recipe categorization, and improved styling to enhance the overall user experience.



Contact

For more projects, visit:

<https://new-portfolio-site-sigma.vercel.app/>

Email: pkernan18@yahoo.com