

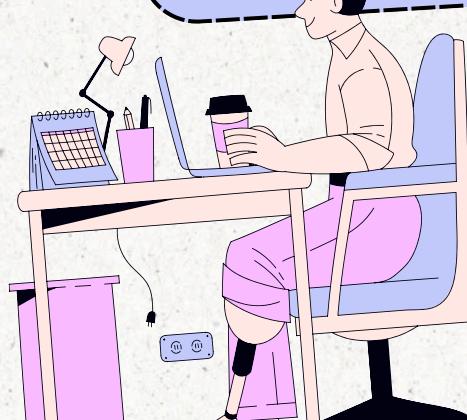
# CASE STUDY PYTHON

## A culinary journey to learning Python Django

**O1**

### Overview

Built and deployed a full-stack recipe application using Python and Django, featuring user authentication, ingredient-based search, and difficulty analytics. Hosted the project on Microsoft Azure.



**O2**

### Purpose & Context

This recipe application was developed as a personal project during my Python Specialization

with CareerFoundry, designed to deepen my

understanding of Python and expand my skills into

web development using Django.



### Objective

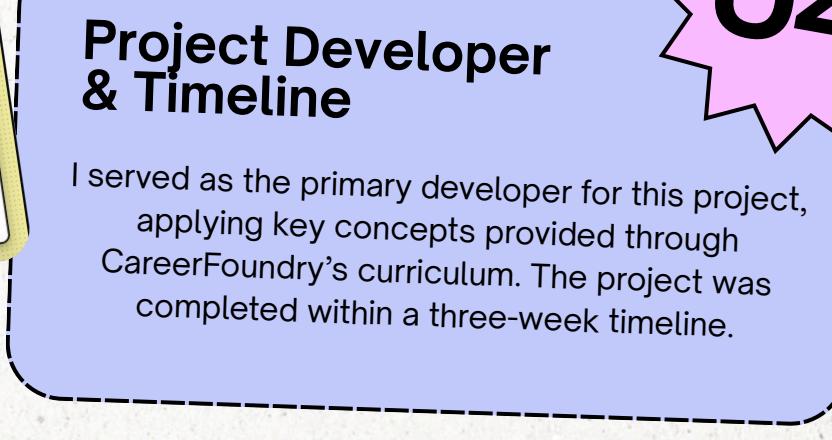
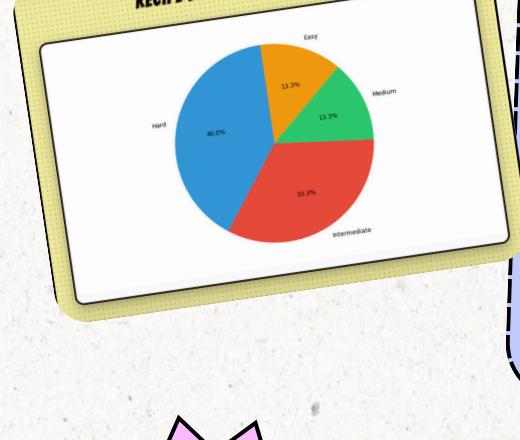
The objective of this project was to create a digital space for food enthusiasts to store and organize their favorite recipes, with features that allow users to search by key ingredients and view recipe difficulty through visualized data.



**O4**

### Project Developer & Timeline

I served as the primary developer for this project, applying key concepts provided through CareerFoundry's curriculum. The project was completed within a three-week timeline.



**O5**

### Technologies

Python | Django | Microsoft Azure |  
Cursor AI | DataFrames | Matplotlib



# CASE STUDY PYTHON

Continued . . .

06

## Setting Up the Project

Setting up a Python application involves many small but essential steps that contribute to the overall development process. One feature I particularly enjoy about Python, compared to other languages, is the use of virtual environments. With a virtual environment, there's no need to worry about system-wide installation conflicts. While it does require an initial setup with a virtual wrapper, once configured, it is simple and low-maintenance.

Another advantage of Python is the ability to run code directly in the terminal. This eliminates the need for a full user interface during testing, which made it easier for me to build and test functionality early on. Before I even started working with Django, I was able to establish a solid project structure through a command-line application.

A major strength of Django is its use of models to define the structure of the entire database. I found the process of creating models to be intuitive and seamless. However, I quickly realized the importance of carefully planning the model structure early on, since making changes later can disrupt associated views and templates.

For detailed setup steps, please see my README file on GitHub: [README file](#)

07

## Views and Templates

The Model-View-Template (MVT) architecture in Django was a new concept for me. At first, the term “template” was a bit confusing since Angular also uses templates, but in a very different way. Once I became familiar with it, however, I found that MVT provided a clear framework for structuring application components, which ultimately helped me better understand Python’s complexity.

Another new concept I encountered was URL routing. Unlike React’s parent-child structure, Python uses routing files to define how different parts of the application connect. I actually enjoyed this aspect—it felt a bit like solving a sudoku puzzle, where I had to carefully track the different moving pieces. While I initially struggled to adjust to this new way of structuring applications, the file system organization helped me stay oriented. Having files labeled as “views” was particularly useful, as it encouraged me to think through each user-facing perspective step by step.

Overall, while MVT presented a learning curve, it became approachable with practice and gave me valuable insight into structuring web applications in Python.

# CASE STUDY PYTHON

Continued . . .

08

## User Authentication

I was impressed by how straightforward Django makes user authentication. Its built-in functionality provides an easy and efficient implementation, which explains why it's such a popular feature among developers. Setting up authentication is as simple as configuring a view and template. While the process for setting up classes and functions differs slightly, it remains just as intuitive.

One area I could expand on is customizing views for individual users. At present, a user can log in with valid credentials, but all users see the same recipes. A next step would be tailoring the content so that each user has a personalized experience.

09

## Data Analysis & Visualization

Data visualization is something I've become excited about, so getting to incorporate it into this project was especially energizing. I've always believed that data is most powerful when it tells a story—or better yet, when it allows the user to feel the story through interaction.

Each recipe was categorized as easy, medium, intermediate, or hard, and then plotted on a graph. This allowed users to quickly identify recipes that matched their skill level and cooking goals. It was a small feature, but it transformed the user's experience by making the application feel more like it was built for chefs like them.

10

## Deployment to Microsoft Azure

Deployment was one of the biggest challenges I faced, as CareerFoundry did not provide specific instructions for implementing Microsoft Azure. Following my instructor's encouragement, I took the initiative to deploy my project to Microsoft Azure. The initial setup was somewhat technical, but I was able to work through it.

The first major issue I encountered was that pages behind the login were not loading. I discovered this was due to the database not being migrated along with the application, which prevented user credentials from being stored. After troubleshooting, I successfully migrated the database, allowing user authentication to function correctly.

The second challenge involved getting images to display properly within the application. The images were being saved with inconsistent naming conventions, which caused broken file paths in the code. Resolving this required several attempts over a few days, but I was ultimately able to correct the issue.

Overall, I am glad I was able to deploy my recipe application to a widely used cloud platform and gain valuable experience navigating deployment challenges.

LIVE WEBSITE DEPLOYED WITH MICROSOFT AZURE

# CASE STUDY PYTHON

## Final Thoughts

11

### Overall Experience

I was excited to have the opportunity to select a Python specialization in my course, with options between Python programming and cloud deployment. Through this project, I was able to gain experience in both areas. Python was a language I had often heard about for its effectiveness and versatility, and this project allowed me to explore its capabilities firsthand.



BLUEBERRY SMOOTHIE

[CLICK HERE FOR RECIPE DETAILS!](#)

12

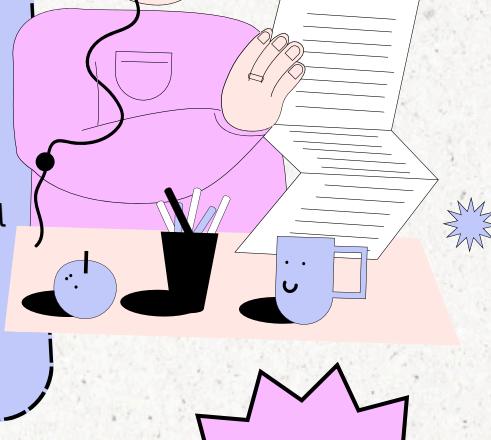
### Learning

Through this Python application, I learned that while new frameworks can feel intimidating at first, they ultimately expand the possibilities for project complexity and user-focused deliverables. I've come to understand that when working with new languages or tools, it's not that I don't know them—it's that I don't know them yet. This mindset has reinforced for me that there is always room for growth and continuous learning.

13

### Challenges

One of the main challenges I faced was entering the project with no prior experience in Django. I had to quickly learn the Model-View-Template (MVT) architecture and adapt to its structure. While defining routes was relatively straightforward, it required careful attention to detail to ensure proper configuration. This process emphasized the importance of precision in declaring components and reinforced the value of writing clean, well-organized code.



14

### The Future

Potential future enhancements include implementing a user sign-up feature to allow individuals to create their own accounts, rather than being assigned access by a superuser. Additionally, expanding the search functionality to categorize recipes by difficulty level (e.g., easy, medium, intermediate, hard) would improve the user experience by enabling meal selection based on skill level.

BOOM!

**YOU HAVE SUCCESSFULLY LOGGED OUT!**

[LOGIN](#)

BYE!