CAREERFOUNDRY

Python for Web Developers Learning Journal

Objective

We find that the students who do particularly well in our courses are those who practice metacognition. Metacognition is the art of thinking about thinking; developing a deeper understanding of your own thought processes. With the help of this Learning Journal, you'll broaden your metacognitive knowledge and skills by reflecting on what you learn in this course.

Thanks to this Learning Journal, when you finish the course you'll have a complete and detailed record of your learning journey and progress over time. We really recommend that you take the time to complete this Journal; students do better in CF courses and in the working world as a result!

Directions

First complete the pre-work section before you start your course. Then, once you've begun learning, take time after each Exercise to return to this Journal and respond to the prompts.

There will be 3 to 5 prompts per Exercise, and we recommend spending about 10 to 15 minutes in total answering them. Don't overthink it—just write whatever comes to mind!

Also make sure that, once you've started filling this document in, you upload it as a deliverable on the platform. This is so that your mentor can also see your Journal and how you're progressing over time. Don't worry though—what you write here won't affect how you're graded for the Exercise tasks. The learning journal is mostly for you and your self-evaluation!

Pre-Work: Before You Start the Course

Reflection questions (to complete before your first mentor call)

1. What experiences have you had with coding and/or programming so far? What other experiences (programming-related or not) have you had that may help you as you progress through this course?
    a. **Nothing formal, I have been around technology as a child as my brother was in IT and I would see him troubleshooting things like viruses and I always thought that was very cool. I got to see a little bit of what coding can do when I briefly did some website designing on no-code/lo code platforms last year. I have always excelled in Math and loved learning calculus and the real world applications of mathematics such as in physics too.**
2. What do you know about Python already? What do you want to know?
    a. **I know it is the main language used in AI, and I want to know how to utilize AI to help create softwares that make peoples every days a little bit easier.**
3. What challenges do you think may come up while you take this course? What will help you face them? Think of specific spaces, people, and times of day of week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.
    a. **I think the time constraints are big. Working full time and sometimes up to 45 hours a week can take a toll on anyone, especially given that free time is meant to**

be used on this course. Finding that right balance to make sure I am still healthy has been a challenge as I love going to the gym but that has to be sacrificed at times. I think coming up with set times that I am supposed to work on my course is helpful.

Remember, you can always refer to Exercise 1.4 of the Orientation course if you're not sure whom to reach out to for help and support.

Exercise 1.1: Getting Started with Python

Learning Goals

- Summarize the uses and benefits of Python for web development
- Prepare your developer environment for programming with Python

Reflection Questions

1. In your own words, what is the difference between frontend and backend web development? If you were hired to work on backend programming for a web application, what kinds of operations would you be working on?
   a. **The frontend is what the user see's and interacts with, it is the code that creates the visual and interactive portions that most people are familiar with. The backend is the connection to the "computer world" that allows one to access all the information that we end up interacting with. I would be working on things such as API's, creating the logic on how data will be extracted (and then shown through the frontend), where things are stored, etc.,**

2. Imagine you're working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option?
   (Hint: refer to the Exercise section "The Benefits of Developing with Python")
   The syntax being very easy to use is a benefit to all of those who work with python. In javascript, the syntax can get convoluted and complicated quickly, and python seems to have that advantage of having a more simplified syntax. It also allows for value reassignment without running into issues which is common in JS, this can be helpful when you're working on large projects since you don't have to troubleshoot into the conflict as much and instead just reassign.

3. Now that you've had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can reflect on the following questions if it helps you. What

do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?

    a. **I want to learn how python is used in AI software development, so that I can continue learning and work with AI as I can see how it is a fast evolving field. I want to work on an actual outside of this course project that reflects my passion for moving into tech, which is to develop a software that unburdens the everyday person. This for me is a software geared towards the healthcare workers who work very hard and could use some competitive edge to heighten patient outcomes and minimize burnout in the field.**

Exercise 1.2: Data Types in Python

Learning Goals

- Explain variables and data types in Python
- Summarize the use of objects in Python
- Create a data structure for your Recipe app

Reflection Questions

1. Imagine you're having a conversation with a future colleague about whether to use the iPython Shell instead of Python's default shell. What reasons would you give to explain the benefits of using the iPython Shell over the default one?

    a. **IPython makes working in Python more comfortable and efficient. It has features like syntax highlighting, auto-completion and descriptive error messages which makes it easier to experiment and debug code. You are able run shell commands directly and has a clear numbered command history. I would also say it helps speed up development.**

2. Python has a host of different data types that allow you to store and organize information. List 4 examples of data types that Python recognizes, briefly define them, and indicate whether they are scalar or non-scalar.

Data type	Definition	Scalar or Non-Scalar?
Integer	Represents whole numbers ie 5, -12	scalar
Float	Represents numbers with decimals ie 3.14	Scalar
List	Ordered collection of items ie ["apple", "banana"]	Non scalar
Dictionary	Stores data as key- value pairs ie {"name": "tea", "time": 5}	Non scalar

3. A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.

    a. **Mutability is the biggest difference between lists and tuples. Lists are mutable, meaning their contents can be changed after creation. Tuples are immutable and cannot be modified once they're defined. The only way to change your previous tuple is to write a new version of your tuple and assign it to the original variable name. Lists are better when you expect changing data, tuples are useful when data should stay constant, which can help prevent accident changes and make code predictable.**

4. In the task for this Exercise, you decided what you thought was the most suitable data structure for storing all the information for a recipe. Now, imagine you're creating a language-learning app that helps users memorize vocabulary through flashcards. Users can input vocabulary words, definitions, and their category (noun, verb, etc.) into the flashcards. They can then quiz themselves by flipping through the flashcards. Think about the necessary data types and what would be the most suitable data structure for this language-learning app. Between tuples, lists, and dictionaries, which would you choose? Think about their respective advantages and limitations, and where flexibility might be useful if you were to continue developing the language-learning app beyond vocabulary memorization.

    a. **For a flashcard based language learning app, I would choose dictionaries stored inside a list. Each flashcard could be a dictionary with a word, its definition and its category for easy access. The outer list would allow users to add, remover, reorder flashcards as the app grows. This structure keeps it flexible for future features like making a quiz feature, or difficulty levels, which would be harder to manage using a structure like tuples.**

Exercise 1.3: Functions and Other Operations in Python

Learning Goals

- Implement conditional statements in Python to determine program flow
- Use loops to reduce time and effort in Python programming
- Write functions to organize Python code

Reflection Questions

1. In this Exercise, you learned how to use **if-elif-else** statements to run different tasks based on conditions that you define. Now practice that skill by writing a script for a simple travel app using an **if-elif-else** statement for the following situation:

    - The script should ask the user where they want to travel.
    - The user's input should be checked for 3 different travel destinations that you define.
    - If the user's input is one of those 3 destinations, the following statement should be printed: "Enjoy your stay in _____!"

- If the user's input is something other than the defined destinations, the following statement should be printed: "Oops, that destination is not currently available."

Write your script here. *(Hint: remember what you learned about indents!)*

```
destination = input("Where would you like to travel?").strip().lower

if destination == "paris":
    print("Enjoy your stay in Paris!")
elif destination == "tokyo":
    print("Enjoy your stay in Tokyo!")
elif destination == "new york":
    print("Enjoy your stay in New York!")
else:
    print("Oops, that destination is not currently available.")
```

2. Imagine you're at a job interview for a Python developer role. The interviewer says "Explain logical operators in Python". Draft how you would respond.
   a. **Logical operators in python are used to combine or compare conditions and help the program decide what to do next. The main ones are "and", "or" and "not". For example, and is used when multiple conditions all need to be true, or is used when only ONE of many conditions need to be true, and not is used to flip a condition from true to false (or vice versa). They're often used in if statements (conditional) to control the flow of a program based on different scenarios.**

3. What are functions in Python? When and why are they useful?
   a. **Functions are like a set of instructions used to perform a specific task. They help organize code and reduce repetition, since you can call a predefined function instead of rewriting the same logic multiple times. This keeps programs shorter and easier to read and maintain. Functions also make code easier to test and debug because they break a program into smaller, focused pieces, allowing you to check one function at a time instead of searching through a long script.**

4. In the section for Exercise 1 in this Learning Journal, you were asked in question 3 to set some goals for yourself while you complete this course. In preparation for your next mentor call, make some notes on how you've progressed towards your goals so far.
   a. **I had a slow start because work and family obligations side-tracked me but I put my foot down and now feel like im powering through it faster than I was before. Im really choosing to leverage AI in helping me understand things rather than struggling through it longer than I should before asking (it might be a pride thing). I also have been doing a lot better at managing my ADHD, given that its been at an all time high and I'm now getting support for that again. I realize the time is now with hiring season being around the corner.**

Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

1. Why is file storage important when you're using Python? What would happen if you didn't store local files?
   a. **It allows data to exist beyond a single run of a script. With no file storage, any data creating during runtime — like user inputs (for example our recipe script) and its outputs, would be lost as soon as the script finishes running. Storing files locally is more realistic since that is how a lot of code works within the real world.**

2. In this Exercise you learned about the pickling process with the **pickle.dump()** method. What are pickles? In which situations would you choose to use pickles and why?
   a. **Pickles are a way to convert python objects (like lists and dictionaries) into a format that can be saved to a file for later use/ testing. I would use pickles when I want to quickly save and load python data say for testing a certain script's logic without having to run the script from the beginning (like in our recipe_input.py). It seems like it would be useful for small projects but not entirely sure how it would translate to more complicated programs.**

3. In Python, what function do you use to find out which directory you're currently in? What if you wanted to change your current working directory?
   a. **I can use tree to see the tree like structure of my folders (downloaded using brew install tree). In python I can use os.getcwd(). If I want to change the current directory, I can use os.chdir() using the path I want to move to. This comes from the os module that makes it easier to manage files when you may be working with someone else's data/code.**

4. Imagine you're working on a Python script and are worried there may be an error in a block of code. How would you approach the situation to prevent the entire script from terminating due to an error?
   a. **I would use try and except blocks, this handles any expected errors gracefully without stopping the script completely. By catching errors I can display messages to the user or allow the script to keep running which improves a scripts stability and is more user friendly.**

5. You're now more than halfway through Achievement 1! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? Feel free to use these notes to guide your next mentor call.

a. **I am proud of the way I have increased my efficiency by meal prepping on my weekends so that I can come home, eat quickly and start working on my projects. I did not realize how taxing and time consuming it was until I added this into my routine. I wish I had practiced it earlier. I also added to my notion calendar (where I add all my obligations) with a block of time dedicated to coding. I was really struggling with how to get started on some of the coding problems, and I began to explore pseudocode and I think it helps me wrap my brain around the code I need to implement. I think having students learn pseudocode or how to start on a new language would be helpful (although I can see why its less useful now with AI). I also realized I work faster if I don't obsess over everything being perfect and understanding everything 100% (90% is pretty good too!). I am trying to practice realizing when there is diminishing returns —- when to move on. This was a struggle I had in the main course and dragged out my timeline tremendously since it became paralyzing at times.**

Exercise 1.5: Object-Oriented Programming in Python

Learning Goals

- Apply object-oriented programming concepts to your Recipe app

Reflection Questions

1. In your own words, what is object-oriented programming? What are the benefits of OOP?
    a. **(OOP) is a way of organizing code using objects that represent real-world things. Each object contains data (what it has) and behavior (what it can do, such as the methods used in this assignment). One major benefit of OOP is that it keeps code organized, since all logic related to an object lives together. It also makes code more reusable, because existing objects and classes can be extended instead of rewritten. OOP helps break larger problems into smaller, manageable pieces that are easier to scale as a project grows.**
2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
    a. **A class is like a blueprint that defines what something should look like and what it can do. An object is an instance created from that class. A real-world example would be a Car class, which defines attributes like brand, model, year, and color, along with behaviors such as starting, stopping, and accelerating. An object created from this class could be a blue 2020 Toyota Corolla. While it follows the same structure and behaviors defined by the Car class, its values are unique because it is its own object.**

3. In your own words, write brief explanations of the following OOP concepts; 100 to 200 words per method is fine.

Method	Description
Inheritance	Inheritance allows a class to reuse and extend another class. A child class inherits attributes and methods from a parent class, which helps reduce duplicate code and makes programs easier to extend. It lets one class build on top of another.
Polymorphism	Polymorphism means different objects can use the same method name but behave differently. This allows you to write flexible code that works with many object types. Same action, different behavior depending on the object.
Operator Overloading	Operator overloading lets you define how built-in operators (like + or Print) work with your own objects, making them easier and more natural to use. It lets custom objects behave like built in types.

Exercise 1.6: Connecting to Databases in Python

Learning Goals

- Create a MySQL database for your Recipe app

Reflection Questions

1. What are databases and what are the advantages of using them?
    a. **They are structures places where data can be stored, organized and used later. They allow you to save save to work with it efficiently. They make it easer to search, update, and manage large amounts of data.**

2. List 3 data types that can be used in MySQL and describe them briefly:

Data type	Definition
INT	Whole numbers, like ID's or quantities or cooking times
VARCHAR	Stores text of a certain character limit like names
FLOAT	Used for numbers with decimals like prices or measurements

3. In what situations would SQLite be a better choice than MySQL?

     a. **SQLite is better for smaller projects, simple apps or learning when you don't need a full database server. Its easy to set up and stores everything in a single file. So projects like a small app or prototyping are better for using SQLite,**

4. Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?
     a. **They honestly feel very different. It feels like Javascript is a more event driven language in the browser and in the depth of user interaction, which makes it great for front end work. Python feels more like a focus on the data handling and backend logic with a more straightforward syntax (for the most part.) Im curious to know the lengths Python is used in todays technology.**

5. Now that you're nearly at the end of Achievement 1, consider what you know about Python so far. What would you say are the limitations of Python as a programming language?
     a. **It doesn't seem like it would be ideal for frontend development and would likely need more frameworks or languages to be able to develop something like Javascript does more naturally. It does have a good beginner friendly feel even if it is not the best tool for every job.**

Exercise 1.7: Finalizing Your Python Program

Learning Goals

- Interact with a database using an object-relational mapper
- Build your final command-line Recipe application

Reflection Questions

1. What is an Object Relational Mapper and what are the advantages of using one?
     a. **An ORM lets you work with a database using Python objects instead of writing raw SQL. You define tables as classes, and the ORM handles the database information behind the scenes. It makes code easier to read, easier to maintain, and less error-prone, especially as projects get bigger**.

2. By this point, you've finished creating your Recipe app. How did it go? What's something in the app that you did well with? If you were to start over, what's something about your app that you would change or improve?
     a. **It went really well overall and helped everything click together. I think I did a good job breaking the app into clear functions and slowly adding features like searching, updating, and deleting recipes. If I started over, I'd plan the structure a bit more upfront and clean up some repeated logic to make the code even more organized.**

3. Imagine you're at a job interview. You're asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.
   a. **I built a command-line recipe app in Python that lets users create, search, update, and delete recipes. I started simple and then added database storage using MySQL and SQLAlchemy. Through that project, I worked with Object Oriented Programing (OOP), databases, Object Relational Mapping (ORMs), and user input validation, and got experience building a full Python app end to end.**

4. You've finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:
   a. What went well during this Achievement?
      i. **I feel like I got a good grasp on the Python syntax. I really liked how we built on the same idea, using the recipe structure and utilizing different ways to build depending on what was needed. It helped demonstrate how there is many ways to run a program using different packages and how it looks/ feels compared to one another.**
   b. What's something you're proud of?
      i. **I am proud of how focused I was able to stay given the holidays being in the middle of everything. I want to keep this momentum going for the next achievement so that I am able to move forward to the job application process.**
   c. What was the most challenging aspect of this Achievement?
      i. **The database part with SQL was the most challenging. When working with MongoDB in our javascript portion I remember feeling challenged by having to learn the syntax and integrate it into our program.**
   d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
      i. **While I do feel like I have a good grasp, I would like to see examples that a developer would be faced with that I could more easily translate, possibly in addition to a project. The resources at the end are helpful to read, however I do recognize that practice projects are the best way to learn.**
   e. What's something you want to keep in mind to help you do your best in Achievement 2?
      i. **That understanding everything perfectly is not always necessary right away because you will be faced with a similar concept at another point. I am frustrated with the amount of time I have spent spinning my wheels on understanding something on my own rather than utilizing AI to help me sooner. I know using it exclusively is not helpful, however.**

Well done—you've now completed the Learning Journal for Achievement 1. As you'll have seen, a little metacognition can go a long way!

Pre-Work: Before You Start Achievement 2

In the final part of the learning journal for Achievement 1, you were asked if there's anything—on reflection—that you'd keep in mind and do similarly or differently during Achievement 2. Think about these questions again:

- Was your study routine effective during Achievement 1? If not, what will you do differently during Achievement 2?
    - **It was effective at the beginning but the holidays did get in the way for a little while. Now with that in the past I think it would be a lot more focused. I have set time aside after I eat dinner to work on it for a few hours after work.**
- Reflect on your learning and project work for Achievement 1. What were you most proud of? How will you repeat or build on this in Achievement 2?
    - **One thing I will repeat is going through the notes a little quicker. In the main part of the program I spent ample hours getting through the notes and jotting things down that I thought were important and noticed my notes got very extensive. I do still want to take good notes, however I want to maximize the time I spend on the actual project.**
- What difficulties did you encounter in the last Achievement? How did you deal with them? How could this experience prepare you for difficulties in Achievement 2?
    - **I actually experienced the least amount of technical difficulties during this portion of the course. However, I did learn to move on or take a break when things feel frustrating, or am experiencing some coding fog.**

Note down your answers and discuss them with your mentor in a call if you like.

Remember that can always refer to Exercise 1.4 of the Orientation course if you're not sure whom to reach out to for help and support.

Exercise 2.1: Getting Started with Django

Learning Goals

- Explain MVT architecture and compare it with MVC
- Summarize Django's benefits and drawbacks
- Install and get started with Django

Reflection Questions

1. Suppose you're a web developer in a company and need to decide if you'll use vanilla (plain) Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?
   a. **Using vanilla python gives you full control and is great for small scripts or simple applications. However you must build everything from the ground up, including routing, security, and database handling, which can be lengthy depending on complexity. Django already has the structure and built in features that are used to speed up development, especially for web apps. The drawback with this is that can can be restrictive (so if you need lots of customization it is not the best choice), and you would have to learn how to use yet another framework.**

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?
   a. **MVT clearly separates responsibilities while keeping things tightly integrated. Django handles the controller logic for you, reducing the boilerplate code needed and allows you to focus on building features right away rather than spending a lot of time wiring all the boilerplate pieces together**

3. Now that you've had an introduction to the Django framework, write down three goals you have for yourself and your learning process during this Achievement. You can reflect on the following questions if it helps:
   ● What do you want to learn about Django?
     ○ **I would like to learn how Django is structured, especially learning how models, views, and templates work together in a real application.**
   ● What do you want to get out of this Achievement?
     ○ **How to set up Django products for data heavy applications. How to work with virtual environments to prepare me for future portfolio projects.,**
   ● Where or what do you see yourself working on after you complete this Achievement?
     ○ **I can see myself building more complex data driven web applications and working towards projects that feel production ready and relevant to real world teams.**

Exercise 2.2: Django Project Set Up

Learning Goals

● Describe the basic structure of a Django project
● Summarize the difference between projects and apps
● Create a Django project and run it locally
● Create a superuser for a Django web application

Reflection Questions

1. Suppose you're in an interview. The interviewer gives you their company's website as an example, asking you to convert the website and its different parts into Django terms. How would you proceed? For this question, you can think about your dream company and look at their website for reference.
   (*Hint: In the Exercise, you saw the example of the CareerFoundry website in the Project and Apps section.*)
   I would start by looking at the different features of the website and mapping each feature to a Django app, similar to how CareerFoundry separates things like login and courses. I'd focus on the pages, user actions, and any data the site manages, then group related functionality into apps. For each app, I'd identify the models (data), views (logic), templates (UI), and URLs (routes).
   In simple terms, I'd be translating the site into what data it contains, who uses it, and what actions those users need to take. For example, in an event-based application, Django apps could include accounts, events, and tickets. Models might include UserProfile, Event, and Venue, while a template could be an event list page and a URL route could be /events. The Django admin site could then be used by staff to create events, edit venues, and manage content without needing to build custom tools right away.

2. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.
   a. **Create and activate a virtual environment using mkvirtualenv <name>**
   b. **Install Django with pip install Django (inside the venv)**
   c. **Create a project using django-admin startproject <projectname>**
   d. **cd <projectname> (be in the folder with manage.py)**
   e. **Run the server using python manage.py runserver to confirm it works**
   f. **create an app using python manage.py startapp <appname>**
   g. **Run python manage.py migrate to set up the initial database tables (Django defaults)**
   h. **When you add/change models later: run python manage.py makemigrations then python manage.py migrate**
   i. **Open http://127.0.0.1:8000/ and confirm it runs**

3. Do some research about the Django admin site and write down how you'd use it during your web application development.
   a. **The admin site is the built in dashboard that allows you to manage your database (Similar to how we used mongoldb atlas previously), without building the pages yet. You would use it in development to create/edit/delete (use CRUD operations), to test that my models are behaving correctly, and manage users/permissions. I would use the admin as a fast way to inspect the data I am expecting while i'm building out the frontend.**

Exercise 2.3: Django Models

Learning Goals

- Discuss Django models, the "M" part of Django's MVT architecture
- Create apps and models representing different parts of your web application
- Write and run automated tests

Reflection Questions

1. Do some research on Django models. In your own words, write down how Django models work and what their benefits are.

2. In your own words, explain why it is crucial to write test cases from the beginning of a project. You can take an example project to explain your answer.

Exercise 2.4: Django Views and Templates

Learning Goals

- Summarize the process of creating views, templates, and URLs
- Explain how the "V" and "T" parts of MVT architecture work
- Create a frontend page for your web application

Reflection Questions

1. Do some research on Django views. In your own words, use an example to explain how Django views work.

2. Imagine you're working on a Django web development project, and you anticipate that you'll have to reuse lots of code in various parts of the project. In this scenario, will you use Django function-based views or class-based views, and why?

3. Read Django's documentation on the Django template language and make some notes on its basics.

Exercise 2.5: Django MVT Revisited

Learning Goals

- Add images to the model and display them on the frontend of your application
- Create complex views with access to the model
- Display records with views and templates

Reflection Questions

1. In your own words, explain Django static files and how Django handles them.

2. Look up the following two Django packages on Django's official documentation and/or other trusted sources. Write a brief description of each.

Package	Description
ListView	
DetailView	

3. You're now more than halfway through Achievement 2! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? You can use these notes to guide your next mentor call.

Exercise 2.6: User Authentication in Django

Learning Goals

- Create authentication for your web application
- Use GET and POST methods
- Password protect your web application's views

1. In your own words, write down the importance of incorporating authentication into an application. You can take an example application to explain your answer.

2. In your own words, explain the steps you should take to create a login for your Django web application.

3. Look up the following three Django functions on Django's official documentation and/or other trusted sources and write a brief description of each.

Function	Description
authenticate()	
redirect()	
include()	

Exercise 2.7: Data Analysis and Visualization in Django

Learning Goals

- Work on elements of two-way communication like creating forms and buttons
- Implement search and visualization (reports/charts) features
- Use QuerySet API, DataFrames (with pandas), and plotting libraries (with matplotlib)

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.

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

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.

Exercise 2.8: Deploying a Django Project

Learning Goals

- Enhance user experience and look and feel of your web application using CSS and JS
- Deploy your Django web application on a web server
- Curate project deliverables for your portfolio

Reflection Questions

1. Explain how you can use CSS and JavaScript in your Django web application.

2. In your own words, explain the steps you'd need to take to deploy your Django web application.

3. (Optional) Connect with a few Django web developers through LinkedIn or any other network. Ask them for their tips on creating a portfolio to showcase Python programming and Django skills. Think about which tips could help you improve your portfolio.

4. You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:
   a. What went well during this Achievement?
   b. What's something you're proud of?
   c. What was the most challenging aspect of this Achievement?
   d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

Well done—you've now completed the Learning Journal for the whole course.