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. I’ve had a few classes in Community College for C++ and Java. Now after Full-Stack Immersion, I have experience in HTML, CSS, JS, and its frameworks, React and Angular. I think that my ability to adapt to varying situations is going to be pivotal in my success in this field and course.
2. What do you know about Python already? What do you want to know?
    a. I know it’s a high-level programming language similar to Java or Javascript. I want to be able to delve more into the back-end languages and understand the mechanics of “why” more than I do now.
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’ve struggled with backend development before with the creation of the movie API so I know that that is where my weakness lies. I will have to focus harder on this part especially since it will be the main focus of Python. I will make sure to set aside more time to this than I did in the Immersion course.

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. Front-end deals with what the user sees. It is everything in relation to the UI and the design aspects that the user might see. It also encompasses the features that the user might be able to use.
    b. Back-end deals with the database, the servers, the data that the user is not able to see, routing, and is the backbone for any great application.

    c. If I were hired for this type of work, I would most likely be dealing with server management, database management, lots of queries for SQL or Non-SQL databases, 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")

A. If I were convinced that Python was the right answer, it would most likely be due to the nature of the project. If the project were to deal more with back-end situations, then Python would be superior. Its readability and built-in packages would have to be weighed in on as well. If the packages that you would need are already in Python, that would make it more desirable.

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 be able to read Python code efficiently without any help.
    b. I want to be able to utilize Python in my future projects.
    c. I want to be able to learn Python to the extent that I can use it as a keypoint when I look for a new job.

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's shell is much easier to use since it visually looks a lot more appealing.

i. iPython uses colors and indentation to make your experience a lot less strenuous on the eye and brain.

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?**
int	Any number without any decimal points.	Scalar
string	An array of characters that can be either alphanumeric or symbols or both.	Non-Scalar
boolean	Either a true or false value	Scalar
list	An ordered sequence that is mutable. Contains various items inside.	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. Lists and tuples are very similar in how they act however, lists are mutable, and tuples are immutable. This means that lists can be modified while tuples cannot (after initialization). Lists use [ ] to surround its units and tuples use ( ).

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. I think that if this language-learning app was going to be a one-to-one translation app, then using a tuple would be best. This way, quick loading and quicker access would be very useful for the usage in the flashcards. However, if this application was going to be a one-to-many situation (where one word/phrase can be translated into many different languages) then a dictionary would be best. This way, you can have an appropriately labeled database. For example, you can have the word "Hello" and its children (in a dictionary) could be labeled, "French", "Spanish", and "Japanese". And their children could be "bonjour", "hola", and "konnichiwa", respectively.

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!)*

```python
travel_destinations = ['Japan', 'Korea', 'China']

user_destination = str(input('Where would you like to go? '))

if(user_destination in travel_destinations):
    print('Enjoy your stay in ' + user_destination)

elif(user_destination not in travel_destinations):
    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 conditional statements that allow you to give branching paths to where your code will go. Without them, your code will flow the exact same way every time, and while that could be useful for super tiny projects, if you want any depth at all you would need to be familiar with Logical Operators.

3. What are functions in Python? When and why are they useful?

    a. Functions are pre-made blocks of code that are able to be called wherever you would like. They are useful when you want your code to be more readable. Instead of having a gigantic block of code that follows a million different pathways, you can instead break it up into smaller, more digestible functions. This also makes it easier in terms of scalability and task reproduction. Functions allow you to create the method once, and reproduce it however many times you would like.

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've learned quite a bit about how Python works. In fact, I was surprised by myself when I was able to write 95% of the code on the first try and had it running correctly. I think the instructions were a bit confusing, but I was still able to write code for the task quite well!
    b. I think that I reached a point where i can at least understand a lot of Python code.
    c. I don't know if I would be comfortable saying it's one of my core skills in an interview yet but I could gladly say I am familiar and am learning it at the moment.

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. Then none of your work would actually be saved. If they are never stored into local files, they cease to exist as soon as your script ends.

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 store data in binary. This means that, to us humans, it is impossible to fully understand and read but to computers, are perfectly legible. This allows us to save data that would be more secure and also more efficient.

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. Os.getcwd()
    b. If you wanted to change your current working directory, you would do: os.chdir(directory_name)

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 make sure to include try and except blocks to try and isolate the errors and instead of terminating, it would go back to before the error.

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'm proud of how I'm able to grasp everything so easily. It hasn't all been 100% for me, I have struggled with the syntax for List Comprehension, and some of the tasks have been worded to seem harder than they are. However, for the most part, I'm enjoying the course and how its been so far!

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 programming framework or model that revolves around the usage of objects and classes. The benefits of OOP range from its reusability, manageability and flexibility.
        i. OOP is reusable because Classes are highly reusable. You can spend the time to create one class that you can create many objects out of with varying functions for different uses. It's manageable because it is very easy to read and understand. Everything is inherently labeled and makes things quite simple to read, even for someone with limited coding knowledge. And it is flexible because you can create new classes using inheritance that can be used for specific situations but still have the methods of the main parent class.

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
    a. Objects derive from Classes. Objects are simply instances of the class that they are based off. Classes define or are the blueprint for the objects that are created. Imagine we have a class called "Teacher". In this class, all teachers have certain attributes, such as a name and a subject they teach. Let's say we have an object named "Brian", who is an instance of the "Teacher" class. When Brian became a teacher, he was initialized with a specific subject. Now, according to the class, Brian, like any other teacher, is capable of

teaching. This is achieved through a method called teach_subject(self, subject), where Brian can impart knowledge on the subject he teaches.

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 in Python allows classes to inherit attributes and methods from their parent classes. For instance, consider the example of a Teacher class mentioned earlier. The Teacher class is a derived class, or child class, while the Worker class acts as the parent class. Both Teacher and Worker classes have a common attribute, "name", but only Teacher has an additional attribute, "subject". Through inheritance, Teacher inherits the method get_payment(self, source, payment) from its parent class Worker. This means that Teacher gains access to the method to receive payment, despite it not being explicitly defined within the Teacher class. This means that Teacher can leverage its functionality gained from the Worker class but also use its specificity due to its status as a derived class.
Polymorphism	In Python, Polymorphism is the ability for derived classes of the same parent class to be able to call on methods from their parent class independently from one another. With two differing classes, both having the “def speak(self)” functions, you would think that one would override the other. However, in Python, due to Polymorphism, both classes are able to independently call on the speak command and it each activating their own definitions of said method. Polymorphism relies on being able to redefine the function itself. For example, let’s say we have the parent class “Worker” and the two derived classes “Teacher” and “Doctor”. Worker has the method get_payment and thus, Teacher and Doctor have the exact same method because they both derive from the Worker class. However, Teacher’s get_payment may return $100, while the Doctor’s get_payment method

	may return $1000. They are the same method, but the redefinition because of Polymorphism allowed it to be used in different ways by different Classes.
Operator Overloading	Operator Overloading in Python refers to the ability to "overload" the functions of built-in operators. This means that operators like “less than” or “equal to” can be redefined to have different behaviors. This capability is useful because sometimes default behaviors may not align with our intentions. For instance, adding two different types together in a print statement might result in an error in Python due to incompatible types. Similarly, a return statement could become convoluted with string concatenations, leading to unclear code. With Operator Overloading, we can redefine operators to suit our needs. For instance, we can redefine the addition operator (+) to concatenate strings or to perform custom arithmetic operations on custom classes. This flexibility allows for clearer and more expressive code. Moreover, Operator Overloading can simplify tasks such as sorting or object comparisons. By redefining comparison operators (like equals or less than), we can customize the behavior of these operations based on the semantics of our classes.

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. Databases are a way for you to manage data entries that can be much too large for your average Excel file to handle. They are useful because they are secure, easily accessible, searchable and can be handled by multiple users. It is a great way to secure your data and becomes mandatory when your business has scaled up to a certain point.

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

Data type	**Definition**

DATETIME	A date type that takes in date time values. Often used for dates such as registration date or date of deletion.
INT	An integer type that takes in any number as long as it does not have any decimals. Used for everything including stock numbers or phone numbers.
VARCHAR	A string type that usually takes in a parameter of how many characters that that item is supposed to have. Often used for things like names or descriptions.

3. In what situations would SQLite be a better choice than MySQL?
    a. SQLite is a lighter way to utilize SQL databases, and such are useful when you are dealing with smaller apps. Especially in-app solutions such as those found in mobile apps or small one-page applications.

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. JavaScript and Python have very different syntax, so much so that looking at one then the other might not seem very familiar at first glance at all. However, you will notice that a lot of the same things apply to both (such as append or sort functions). JavaScript and Python differ in their functions especially in their primary functions. JS is commonly used in front-end and some back-end applications, while Python is the opposite: it is commonly used for back-end programming. Python also seems to be much simpler to understand when looking at it since it uses very simple syntax and understandable keywords (ex: for loops).

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. To be honest, I'm not entirely sure what the limitations are since I feel like I've only been scratching the surface so far. So far though, I can infer that perhaps Python might be a bit slower than other compiled languages like C++. It is also quite a bit different when it comes to data visualization. It is easier and better to connect it to a database and run things on the CLI but showing things on the browser might be more difficult.

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 Object Relational Mapper is a tool that is used to automatically “translate” or help you in your database conversion process. Due to there being many versions of SQL databases, they often have slightly (or vastly) different syntaxes that can cause issues inbetween conversions! This also means that learning one database could have or could not have transferrable knowledge (syntax-wise, they should operate similarly enough that teaching one will help you learn others). An ORM makes this easier for you by managing the syntax side on its own so you only must worry about your actual Python (or other programming language) code.

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. I honestly really enjoyed creating this app and it didn’t take as long as I thought it would. It took me about a week from start to finish for Achievement 1, which feels shorter than others, but I think a lot of that was due to the time I spent, the enjoyment I had going through it and the ease of instructions in the course. I really enjoyed the varying ways the app could go and the potential for these types of applications. I’m surprised by how easy the syntax is for Python also! If I had to start over, I would make sure I’m utilizing more functions since my code does look quite cluttered and the usage of methods and functions would allow it to look a lot more readable.

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. Python was the most recent programming language I learned but it certainly wasn’t the least. While working on a project where I created a Recipe App where you could create your own recipes, view or search for them, and even update or delete recipes, it dawned on me the power of Python. The vast capabilities and ease of understanding that the syntax offered was something I thoroughly enjoyed. I realized that back-end coding could be genuinely interesting and fun and that I would be open to learning more about Python and other high-level programming languages in the future. While working on the Recipe project, I oft found myself hours away just toiling away, bug fixing and feature adding until I realized that the sun went down and up again. It was a wonderful experience and one that I’ll always remember fondly and look forward to revisiting in the future.

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 think that I learned a lot about how Python works, the syntax of it all and I now see the potential in this language!
    b. What’s something you’re proud of?
        i. I’m proud of the limited number of times I had to look things up. Python syntax is very easy to follow and understand and that made things very streamlined when I wanted to add extra functions that the course did not ask for (which I do in all of my CareerFoundry assignments).
    c. What was the most challenging aspect of this Achievement?

        i. I think the most challenging aspect of it was getting around the syntax for the ORM or just SQL in general. It isn't something I'm TOO comfortable with but I definitely feel a lot better with after this Achievement.

d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
    i. It surpassed my expectations greatly! To be honest, I was expecting this to be a boring Achievement (no offense!) where I learned a lot of back-end work. However, while I did that, I realized that it wasn't boring at all! This was one of my favorite Achievements in my entire time here in CareerFoundry.

e. What's something you want to keep in mind to help you do your best in Achievement 2?
    i. I want to make sure that I'm keeping myself accountable so I can finish strong in my last technical course in CareerFoundry (still must finish Job Prep!).

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?
    - My study routine was very effective! I was able to learn a lot about Python in a very short amount of time and I retained quite a bit of it.
- 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?
    - I was most proud of how quickly I was able to grasp the content. Python is very easy to read and the flow of it became very clear to me.
- 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 had difficulty making clean and concise code. I often found myself forcing solutions to work instead of figuring out proper ways to write my code. I want to work on writing more functions to clean my code up and make it more streamlined in the process.

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. The advantages of using Django are prevalent early on. It has a robust system that allows for much easier Pythonic code. Instead of dealing heavily with the intricacies of raw Python, you would be able to use functionalities inherent in Django and its vast plethora of built-in plugins. It also has fast deployment and processing to make sure that your application runs efficiently.
    b. However, on the downside: Django is extremely strict on how it is structured. So much so that your code will need to be structured the way that Django calls for it. Otherwise, things may not work. Django's built in plugins may also play a hindering part when you realize that ALL of it comes with Django, even parts you may not want to use. This means that unused code can still slow down your application. If your application is small and doesn't use a database for example, you would probably not want to use Django.

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 is easier to use and understand because it does a lot of the back-end coding for you. MVT is also simpler because it's primarily controlled by the View. In MVT, the Template controls a lot of the functions. So much so that Django comes with its own built-in ORM.

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 want to learn how to deliver an app from Concept to Production
    - What do you want to get out of this Achievement?
        - I want to be able to be confident enough in Django that I can talk about it like I can React or CSS
    - Where or what do you see yourself working on after you complete this Achievement?
        - I see myself as being able to meet all my goals and adding another high-quality project to my portfolio.

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 describe how each of its different functions can be written as a django "app" of the website, while the data that is being used and drawn from the database is the Database, controlled by the configuration files. Each major function of the web Application, including but not limited to, the log-in, registrations, different displays for different datasets, and different pages/views all count as different Django apps. These may be made specifically for the web application that was created, or be taken from other Django applications and shared with the current web application.

2. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.
   a. To begin, set up a clean virtual environment specifically for your Django application. Install Django and create your project using Django's startproject command. Activate the virtual environment, apply migrations with migrate, and start the development server with runserver to run your application. From there you can create your super-use for admin purposes.

3. Do some research about the Django admin site and write down how you'd use it during your web application development.
   a. It would be most beneficial for storing different users and applying different roles/abilities to each user. If you don't want a user to see things, you would be able to limit their control here. This would also be useful when you need to delete users or add them in manually.

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.
    a. Django Models serve as representations for the Django framework. They serve as a way for users and developers to interact with the back-end from the front-end. They allow users to utilize the data stored in the servers and incorporate the Model portion of the MVT structure.

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.
    a. It is very important to write test cases because, at the end of the day, we are human and we are unable to check for absolutely everything in applications that could be very large. When dealing with a small application, there is little need to test everything with code since you can always test it manually. However, once you are in a large codebase with multiple developers and multiple different functions interacting with one another, then test cases are extremely important. For example, let’s say we have a large database with dozens if not hundreds of different functions and models working together to make a function application. One day, a developer decides to add in a new feature that suddenly breaks everything else. Little did he know that the function he created affected another function somewhere else entirely and that caused everything to fail. If the codebase did not have any proper test cases, it might be very difficult to find exactly where and why the error is occurring.

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.
    a. In Django, the views are what is responsible for interacting with the back-end. It directly correlates between the user’s actions and the application itself, acting as a sort of middle-man between the Template and the Model. For example, in order to figure out which Template to show the user, and which Model to draw information from, the view takes in the user’s actions in order to display and use the right elements.

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?
    a. I would use Class-Based views because of their emphasis on reusability. What they lack in customizability, they make up for in being easy to reuse and reapply in different contexts. The same class-based view can be equipped with different methods to allow it to act on different situations that may or may not appeal to different views.

3. Read Django’s documentation on the Django template language and make some notes on its basics.
    a. Django uses 2 different kinds of views:
        i. Class-Based which are more useful for reusability and Function-based which are more useful for ease of use and customizability.
        ii. Templates define what the users see
        iii. Models define what data is used.
        iv. Use urls.py to define your urls
        v. Use views.py to define your different views
        vi. Views take into account user input to decide on which template to show to the user, then use the models to decide on which data to present in the template.

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.
    a. Django’s Static files are files that do not change based on how the user interacts with them. These files stay the same as per the developer. This includes assets like images or gifs, CSS files that dictate the view of the page, JavaScript that define how actions happen in the application, etc;

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	The view used to display multiple instances of a table or tables. Used when you want to display a list or collection retrieved from a database.
DetailView	The view used to display one instance of a table. Used to look into specific entries and display accurate information based on the developer’s code.

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.
    a. I think that I’m personally struggling with Tests and writing accurate tests. As of right now, it has been difficult for me to come up with my own tests and create ones that would benefit my application. I’ve tried GOogling ways to write tests (for instance, in the pressing of buttons) but nothing has made too much sense on it so far? I am hoping that I will learn more about it in the future. More of the syntax, less the actual logic.
    b. Actually syntax is a big issue for me with Django. It doesn’t seem very intuitive in the way that the code is written. Things like {% object of object %} just doesn’t sit right with me to be honest. I know I’ll get used to it in the future but as of right now, it isn’t comfortable with me yet. Other syntax like those for the urlpatterns or settings.py are going to be difficult to understand or memorize. Aside from that, I’ve thoroughly enjoyed making the application and think I’ve made some fun stuff to it!

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

Reflection Questions

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.
    a. Without authentication, then you would have anybody able to access anyone’s accounts. This is not only a privacy issue but it also creates terrible continuity errors. That means that if the application has any sort of user-specific functions like a favorites list or even usernames, then none of that will be the same the next time you log-in. This would also cause issues because multiple users could affect the same functions at the same time, not even talking about the strain this would cause on your application. So if you want any sort of uniqueness in user experience, one where user’s can cater and create their page to fit what they want to do, then authentication needs to be enabled. Users need to be able to feel safe as well that their passwords are kept safe by the application developer’s code.

2. In your own words, explain the steps you should take to create a login for your Django web application.
    a. FIrst and foremost you need to make sure that you create your views for your login and log out. Import your specific plugins for authenticating with Django at the top of the page before anything. Function based views are best for this since you would need to be able to do a very specific task that needs to be highly customizable since you are dealing with http requests. Then you would connect these views to your application’s urls. After that you can update your settings.py to incorporate these new paths. Finally, you must create your templates and their corresponding CSS files to finalize the log-in functions.

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()	Authenticate is Django’s built-in function for verifying the identity of a user. It comes with

	pre-built form validation as well so you don't need to worry about having usernames or passwords that don't align with each other. authenticate() verifies that the combination of username and password is one that exists
redirect()	Redirect is used for when a user has not gained access to a specific page so then it is redirected to another - in this case it was a log-in page. This can also be used to restrict access for places such as an admin page or moderator page.
include()	Include is used to connect the views and urls from different apps in your application. We used it in the **Recipe App** by connecting the **recipe** app to the **recipe_app** app. From there we were able to use the .views of **recipe** inside of the urls in **recipe_app**

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.
    a. On Youtube, the data that could be collected from me could be things like my country of origin, my age-group, my likes or dislikes or even just my watch history in general. All of these things could feed into my algorithm to give me recommendations to watch that would suit my interests and would be appropriate for me to show. If someone is underaged, for example, you wouldn't want to show them violent or sexual content.

However, if there is someone who has watched quite a bit of that sort of content, and is over the age, the application can push those types of suggestions forward.

2. Read the Django official documentation on QuerySet API. Note down the different ways in which you can evaluate a QuerySet.
    a. A QuerySet can be iterated over and over again either normally or asynchronously.
    b. A QuerySet can also be sliced to obtain the desired data.
    c. A QuerySet can also be Pickled or Cached to hide or read data.

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.
    a. QuerySets deal directly with the database itself, allowing for more precise control over your database. However, when you want to actually analyze that data, QuerySets fall short.
    b. DataFrames allow you a greater range of possibilities with which you can manipulate your data. This means greater usage of filters, grouping, or joining together. This makes it the clear winner when it comes to data analysis.

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.