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?

Answer: I have completed the Full-Stack Immersion part of the Web Development program with Career Foundry. Prior to that, I did not have much experience at all. I had done a month of the FreeCodeCamp program which is what sparked my interest in programming and led me consequently to CareerFoundry.

2. What do you know about Python already? What do you want to know?

Answer: Besides the fact that it is a programming language and a very popular one, I don't know anything about it.

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.

Answer: I believe keeping myself motivated may be a challenge as the content of the course can sometimes present us with very complex ideas. I try to keep myself motivated by thinking that in a few years, I will be glad I started right now. I also try to think of the possibilities that completing this course will bring me and what possible doors might open when I can finally and confidently say that I am a web developer. But most of all, I know the importance of breaks and coming back to the course with a clear mind.

EXERCISE 1.1

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?

Answer: To put it shortly: Frontend is what users interact with on an application or web-page, while the backend is what happens backstage and makes the program work. In other words, all of the work that users don't need to see in order to interact with the platform.

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?

Answer: The similarity between JavaScript and Python is that both are object oriented (which means they both work with sets of code and data). Also, both work with variables and functions and give the developer the ability to work with classes and properties. On the other hand, the main differences lie on:

- Speed: where JavaScript tends to perform faster,,
- Syntax: with Python offering a simpler syntax than JavaScript.

When it comes to convincing my team to use Python, I might argue that Python tends to be more suitable for large-scale projects and that working with Python for backend allows us to have versatility due to its very simple language in terms of syntax. I might also add that there are a lot of different frameworks available and resources on how to work with python; besides a strong and collaborative community online.

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 reect 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?

Answer:

1. I want to learn enough about Python so that I can work with it and discuss it confidently with my future co-workers. I want to be able to understand and follow guidelines on possible Python documentations I may come across.
2. By working on this achievement I hope to get a good introduction about Python so that I can grow my confidence in web development and reach my goal in point number one.
3. After the completion of this achievement, I see myself applying for Technical Writing jobs that will allow me to grow my knowledge even more. I also hope to work on smaller projects for myself and my portfolio using the skills that I have acquired throughout the course.

EXERCISE 1.2

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?

Answer: To put it shortly, the iPython Shell is more practical and user-friendly than the default one. IPython's syntax highlighting makes it easier to read the code and it indents the code automatically. IPython Shell is also faster when it comes to testing small pieces of code.

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	Represents integers, includes both negative and non-negative numbers.	Scalar
float	Holds decimal numbers and includes both negative and non-negative decimal numbers.	Scalar
bool	Represents boolean where data can only be either True or False.	Scalar
dictionary	Unordered set of items that requires a key-value pair.	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.

Answer: The difference between Lists and Tuples lies in the fact that lists are mutable and tuples are not. Elements within a list can be modified, deleted or rearranged.
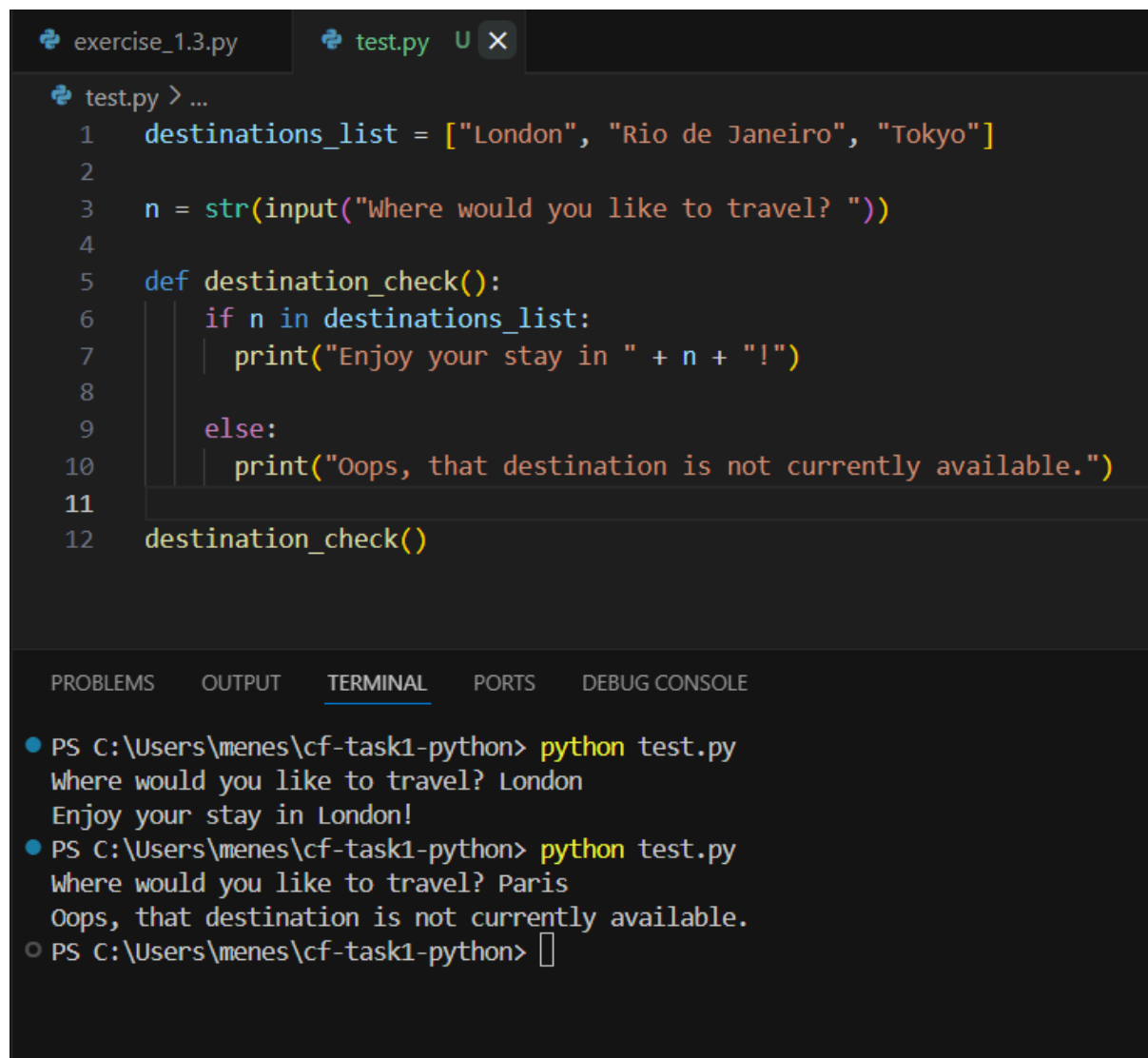
4. In the task for this Exercise, you decided what you thought was the most suitable data structure for storing all the information for a recipe. Now, imagine you're creating a language-learning app that helps users memorize vocabulary through flashcards. Users can input vocabulary words, definitions, 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.

Answer: I would choose to go with dictionaries as they don't require data to be sequential. This particular feature of dictionaries may come in handy as we continue to develop and grow the app's data.

EXERCISE 1.3

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."**

```python
destinations_list = ["London", "Rio de Janeiro", "Tokyo"]

n = str(input("Where would you like to travel? "))

def destination_check():
    if n in destinations_list:
      print("Enjoy your stay in " + n + "!")

    else:
      print("Oops, that destination is not currently available.")

destination_check()
```

```
PROBLEMS    OUTPUT    TERMINAL    PORTS    DEBUG CONSOLE

PS C:\Users\menes\cf-task1-python> python test.py
Where would you like to travel? London
Enjoy your stay in London!
PS C:\Users\menes\cf-task1-python> python test.py
Where would you like to travel? Paris
Oops, that destination is not currently available.
PS C:\Users\menes\cf-task1-python>
```

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.

Answer: Logical operators in Python help you compare conditions in data. The operators are "and", "or" and "not", each having its own function when checking conditions that result in either a True or False response.

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

Answer: Functions in Python work as a set of instructions for the code to perform a task. Python has a set of predefined functions available, but developers can also write their own function. They are useful because they help maintain a clean and concise code, which is easier and faster to work with. They should be used when the code repeats itself, so by using function, your code will be shorter and more condensed.

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.

Answer: I would say I have made good progress towards my goals. I have learned a lot about Python so far and I am slowly feeling more comfortable with this new programming language. I look forward to learning more about this language and how it compares to other languages I know, such as JavaScript.

EXERCISE 1.4

1. Why is file storage important when you're using Python? What would happen if you didn't store local files?

Answer: File storage is important because it allows us to permanently store data in our machines, as well as keeping track of values.If you didn't store local files, the data would be lost when the script stops running and couldn't be retrieved again.

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?

Answer: Pickles "convert complex data into a packaged stream of bytes". You would use pickles when you have a text file with complex structures (such as dictionaries), which is difficult to store in regular text - pickles convert this complex data and store it in a binary file for machines to read.

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?

Answer: To find out which directory you're currently working in, you can use the `os.getcwd()` command. To change the directory in which you're currently working in, you can use the `os.chdir()` command.

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?

Answer: I would use the `try-except` blocks to prevent my entire script from terminating.

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.

Answer: I think the course is going well. I can tell it is getting more complex and I can potentially say I am moving past the basics with this programming language. I'm proud that I

have been able to understand the course and have managed to get multiple tasks approved. I believe I struggle with getting tasks started, but once I go back into the course and revise what I have just read, the instructions become clearer. I also believe I need more practice with organizing my initial thoughts on how to approach a task from the beginning.

EXERCISE 1.5

1. In your own words, what is object-oriented programming? What are the benefits of OOP?

Answer: Object-oriented programming (OOP) is a concept within web development where code is organized in chunks (called objects) within "classes" that can be reused; in other words, code is inherited from previously written code. The main benefit of OOP is that it avoids repetition of code and it therefore allows for a cleaner, more concise coding structure - as well as allowing a more flexible maintenance of such code (aka debugging).

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.

Answer: As mentioned in the course material: "Everything in Python is an object". Objects contain data that is stored in attributes, while classes serve as templates with instructions on how to create objects. Classes define the structure of objects and contain methods and functions that dictate the object. A real-world example could be a bouquet of flowers, where classes would specify which flowers to use and how to arrange them, while the object would be the bouquet itself.

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 programming is when methods of a certain class can be reused in another without the need to re-write such methods. This avoids code repetition and makes for a cleaner and more concise coding structure. When dealing with inheritance, there is always a "subclass" or "inherited class" that uses data from the "parent class" or "base class".
Polymorphism	Polymorphism is when different objects have the same name but their performance is different according to where this method is defined within your code. The perfect example to illustrate this concept in polymorphism is the len() method, which returns different results depending on the data type it is used with but possesses the same name at the end of the day.
Operator Overloading	Operator overloading is the process of defining methods for already existing operators such as "+" or "-". Much like the len() method mentioned above, operators can perform differently depending on which data type they are manipulating. When working with operator overloading, you need to "define a function that Python already reserves for an operator" - these function definitions are then surrounded by double underscores (e.g. __add__).

EXERCISE 1.6

1. What are databases and what are the advantages of using them?

Answer: To put it shortly, databases are stored, organized collections of data. One of the advantages of using databases is their standardized format which can be accessed and formatted more easily than data stored locally. Another advantage is the ability to add a password access to keep your data safe.

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

Data Type	Definition
DATETIME	Holds date and time values.
FLOAT	Holds decimal numbers and includes both negative and non-negative decimal numbers.
INT	Represents integers, includes both negative and non-negative numbers.

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

Answer: SQLite would be a better option than MySQL when you're working with very simple databases, like storing email addresses. It might also be a better choice if you just want to test a database instead of setting up an entire database engine.

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?

Answer: When looking back at the immersion course, I can tell that the differences between Python and JavaScript are substantial. Python seems easier to read, especially when you're dealing with very long scripts and files. Python also feels more modern and more up to date due to its simpler syntax, to the point that it almost seems intuitive. Both are very complex languages however, and require a lot of practice and research when trying to master them.

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?

Answer: I have noticed that one of Python's limitations may be the indentation-oriented syntax, which is easy to mess up if you're not focused. Luckily, VSCode helps you keep track of this feature. Python also seems to be more backend oriented so far, but I must admit I haven't worked with it in frontend so my opinion may change in the future.

EXERCISE 1.7

1. What is an Object Relational Mapper and what are the advantages of using one?

Answer: An Object Relational Mapper (ORM) allows the developer to set the SQL syntax aside and work only and directly through classes and objects, besides saving the developer a lot of time, it makes the coding process easier.

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?

Answer: I believe the process went well, despite encountering many bugs on the way - this leads me to what I did well: I managed to fix all of the issues I came across during the development process and successfully made the app work. If I were to start over, I would like to change how the data is displayed in the terminal, improving how the print() methods show all of the information to the user.

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.

Answer: I would answer this question by saying that I have experience in creating a recipe app with Python, that allows the user to create, edit, view and delete recipes from an app. This allowed me to get some knowledge around how SQL syntaxes work, as well as understanding the importance and value of ORM. Also, this app gave me an extensive introduction to the world of Python that I will for sure put into good use in later projects.

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?

      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 Python skills?

      e. What's something you want to keep in mind to help you do your best in Achievement 2?

Answer: I'm extremely proud of myself for completing this achievement. There was a lot of information to take in and I managed to push through all of the challenging bug-fixes I encountered. The achievement met my expectations and the content in the course was well-written. In terms of confidence, I believe I still have to practice what I've learned but I'm sure that this confidence will come with time and experience. For the next achievement, I'd like to keep in mind that I am able to fix any issues I encounter throughout the course.

EXERCISE 2.1

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?

Answer: The main advantages of Django include its fast and easy development process, its fast processing and high speed, its well-known scalability benefits, its DRY principle which keeps your code non-repetitive or redundant, and lastly its extensive support and community online which allows you to overcome problems you may encounter in the development process. On the other hand, some of Django's drawbacks include its firm structure which doesn't leave much room for the developer to diverge from Django's rules and Django's server intensive nature which increases the complexity of your project.

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?

Answer: The most significant advantage would be that you don't have to write code to fetch data from a database and map it to the URL. The framework deals with the preparation and sending of such data.

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?**

● **What do you want to get out of this Achievement?**

● **Where or what do you see yourself working on after you complete this Achievement?**

Answer: I would like to see Django's advantages and benefits being put to practice. I would also like to acquire a deeper understanding of how frameworks affect the development process and more specifically how Django can improve such processes. Lastly, I hope that by the end of this Achievement I will be able to confidently choose what frameworks are more suitable for my projects.

EXERCISE 2.2

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.)

Answer: For this question, I'll be looking into Readymag's website (https://readymag.com/), where the whole website would be considered the *project*. Login, join, solutions, pricing, examples, templates and learn would all be apps.

2. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.

Answer: Before deploying your Django application, you first need to create a virtual environment. This virtual environment then needs to be activated with the command `activate.bat` so you can create your project by running another command, `django-admin.exe startproject <name of the project>.` Then you would run migrations to create your database (`py manage.py migrate`) and run server (`py manage.py runserver`) to deploy your application and see it in the browser. Note that these commands are Windows OS, if you're using another OS the commands might be different.

3. Do some research about the Django admin site and write down how you'd use it during your web application development.

Answer: The Django admin site is a built-in interface for managing database content in a web application, so I would use it for this very purpose: managing my database. This interface would also allow me to perform CRUD (Create, Read, Update, Delete) operations and to manage users and permissions. I might also use this interface to help me with testing, debugging, and quick data entry. All of this would save me some development time.

EXERCISE 2.3

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

Answer: Django models are objects in Python that the framework uses to access and manage data from the database. These models also define the structure of such data. The main benefit of working this way is that Django "communicates" with the database in a much simpler way than SQL for example. With Django models, the developer only needs to write the structure and the framework deals with the communication with the database.

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.

Answer: The first thing that comes to my mind when I think of why testing is crucial is: to make your life as a developer easier. Testing allows you to encounter problems and bugs in your code early on and fixing these issues as early as possible lets you save time and avoid headaches in the future. Besides, testing makes for a more trustworthy code, especially when working with multiple team members.

EXERCISE 2.4

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

Answer: To put it briefly, Django views work by receiving requests from the user in the browser and returning responses. You can take the following url as an example: exampleurl.com/home. Django selects the right view to load to the user based on this URL coming from the browser, which in this case should present the user with the homescreen of this fictional website.

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?

Answer: Considering that I will have to reuse lots of code in various parts of the project, I would use Django class-based views (CBV). This method provides a way to create views that is easier to reuse and extend when compared to function-based views (FBV), and will help me avoid code repetition.

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

Answer:

- The Django Template Language (DTL) is used to dynamically generate HTML content in Django templates.
- It allows developers to include logic in templates without writing Python code directly.
- *Variables* allow you to display dynamic data in templates.
- *Filters* modify the output of variables.
- *Tags* control the flow of logic in templates.
- *Template Inheritance* helps avoid code repetition by allowing templates to extend a base template
- DTL keeps the logic in views and templates clean, making the code more maintainable and readable.

EXERCISE 2.5

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

Answer: Static files are those that are not automatically generated by Django, such as CSS files, Image files or JavaScript files. These files need to be stored in specific locations and addressed with a corresponding path so that Django can access them properly.

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	ListView is an automated feature in Django that "paginates" the list in context. Meaning, when using this feature, the objects in the list will be displayed in the browser as an actual list.
DetailView	DetailView is another automated feature in Django that takes the attributes' data of a certain object and displays or "paginates" it in a browser through its correspondent path.

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.

Answer: I am very proud of myself for coming this far, specially with being consistent in my learning routine and being able to meet the time restrictions presented by the course. I am struggling with having a positive attitude towards my next steps in the course (the job preparation course), since I have heard from industry experts that it might be hard getting a Junior position at the moment. I would most definitely like to practice all aspects of the course, from the immersion to the elective.

FRONTEND INSPIRATIONS
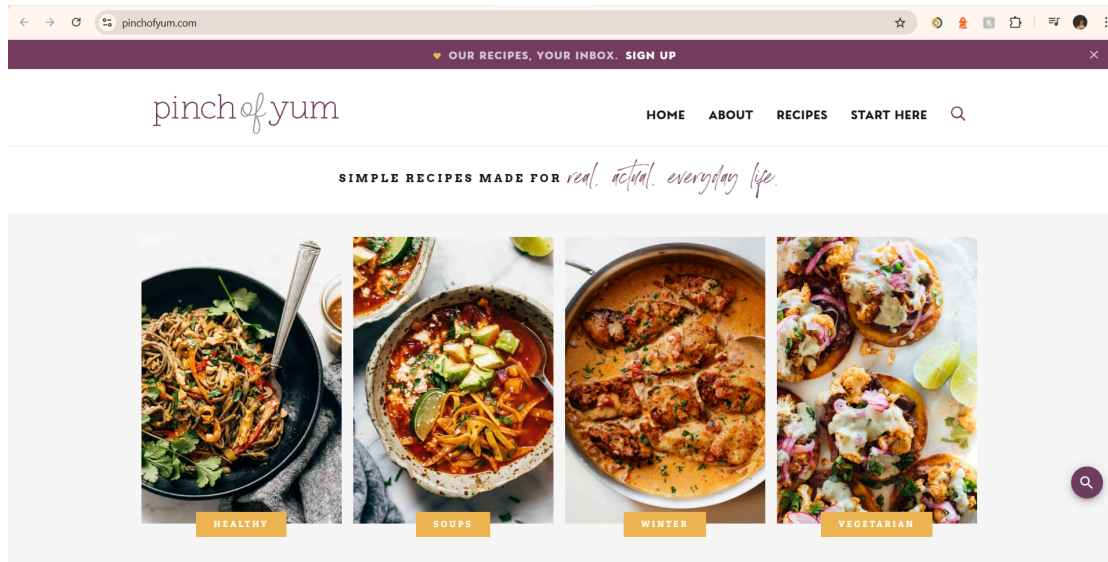
Based on the screenshot above, I gathered some inspiration on how to design the list of recipes for my application. I will try to replicate the rectangular pictures, positioned side by side, with my own recipes.

EXERCISE 2.6

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.

Answer: The main importance of incorporating authentication into an application is for security and protection. Authentication allows users to only access pages that they are actually authorized to see and manage. This also allows for a more personalized experience for the application, where each user can have a unique view when logging in.

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

Answer:

1. Make sure you have a user registered in you /admin.

2. Create a login view file in your src folder.

3. Write a function login_view in this file, making sure to import the necessary libraries.

4. Create a template for the login_view.

5. Register the URL in the project's urls.py file `(path('login/', login_view, name='login'),` make sure to import the view.

6. Run the server to check if it works and link it to an html if so desired.

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( )	Verifies credentials and returns a user object if the credentials are valid. It's commonly used in the login process to validate username and password.
redirect( )	Returns an HTTP redirect to a specific URL which could be a view, model or another URL.
include( )	Includes other URLconf modules in the project's URLpatterns. It keeps the project's URLs structured and organized and it's used for referencing other URL configurations.

EXERCISE 2.7

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.

Answer: I will take the website https://filmow.com/ into consideration for this analysis. In this application, the collected data can be "Films Watched", "Want to Watch", "Don't Want To Watch", "Favorite Movies", "Favorite Actors", "Star Reviews" and so on. Analyzing this data allows them to suggest movies in a more efficient manner based on the data they have on the user. They also have a compatibility feature where, when viewing someone else's profile, you can see how compatible your taste is based on the reviews the user has previously made. By analyzing this data, they can also make better recommendations of movie lists/folders, where users can find a wider variety of movies.

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

Iteration – When you iterate over a QuerySet (e.g., using a `for` loop), Django fetches the records from the database.

Slicing – Accessing a subset of records using slicing triggers evaluation.

Boolean Evaluation – When used in a boolean context, a QuerySet evaluates whether it contains any results.

Using `list()` – Converting a QuerySet to a list forces evaluation.

Calling `len()` – This executes the QuerySet and counts the number of records.

Aggregations – When you use functions like .count(), .sum(), .avg(), Django evaluates the QuerySet.

Fetching a Single Object (.get(), .first(), .last()) – These methods trigger execution immediately.

Explicitly Fetching All Data (.all()) – Though .all() itself does not evaluate the QuerySet, using it in another evaluation method does.

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.

Answer: The advantages of using QuerySet are related to database optimization through ORM, security with built in authorization, and a lazy evaluation which mean QuerySets are only called when needed (this makes for better performance). The disadvantages of QuerySets rely around data analysis, where QuerySets are less flexible in terms of statistical or analytical functions, database dependency, where the performance is tied to a database which is slower than in-memory.

The advantages of DataFrame include powerful data manipulation, filtering and transformation, in-memory computation which is stored in RAM. The latter is also a disadvantage since it can consume a lot of space, alongside the lack of built-in security. DataFrame is better for data processing because it has faster in memory computation and better and powerful analytical analyses.

EXERCISE 2.8

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

Answer: CSS and Javascript can be used in Django web applications in three ways: Inline (where code is written in the style attribute inside the element in HTML), Internal (where a script tag with the code is placed on the head part of HTML) and External (where static files are created and placed in their proper folders, along with specifying their paths).

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

Answer: The first step to deploy is to have up-to-date code pushed in GitHub, you then create a Procfile with the appropriate content and place it in its respective folder. Install gunicorn and dj-database-url, then configure the database in settings.py. Install WhiteNoise and update your settings.py. Change DEBUG to false and hide your secret keys. Perform pip freeze and obtain your requirements.txt file. Upload your website to heroku by creating an application and uploading it, set the database table and superuser and visit it in the browser.

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?

Answer: I think the course content is really well written, and understanding and following the tasks went well. I'm proud to have finally reached the end of the course and to have successfully gotten my tasks approved. Perseverance was challenging since the course took a long time and there is so much to learn, but I pushed through and managed to finish it successfully. I believe this Achievement exceeded my expectations, it turned out to be more straightforward than I had expected and my confidence has definitely grown.