Python – CF – Learning Journal

1.1: Getting Started with Python

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?

Frontend web development is related to development of programs that are interacting directly with the User through the Web browser (Chrome, Safari etc…). Resulting Apps are executed by these browsers and rely on HTML, CSS and JavaScript.

Backend web development on the other hand is related to some code that is run on a server instead of a web browser. It is usually used for more complex programs that often are interconnected with a database. Main backend languages include JavaScript, C, C++, Ruby etc…

If I was hired to work on backend programming for web app, I would like to work on operations that make use of API and databases that make use of mongoDB technology (or similar NoSQL technology) as I found it very flexible technology (scale up, adjust 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")

Python is easy to learn and understand. Like JavaScript, it is a scripting language however from the beginning it was developed with the idea of a quick development in mind in particular with a great readability (indentation to separate blocks of code for instance). Simple built in package management as well as strong community support are also great benefits that support the development of the App using Python.

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?

- I would like to learn how to develop a full stack App using Python (Front and Back end) as we did using Node.js in order to get to understand better on real project the advantages of each languages
- I would also like to learn more about Python as I understand it can be used for data science
- In the future, I believe Python would surely be an option for some of the Apps I would be working on and therefore I would need to be able to contribute to the discussion around the choice of Python or another language

1.2: Data Types in Python

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?

iPython shell is much more **user friendly**, in particular when it comes to **reading the code** (it will display different features of the code in contrasting fonts and colors) but also, iPython Shell will display some **guidance** (in particular a suggestion of possible methods that can be applied to the program variables for instance).

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	This data type represents integers, including both negative and non-negative numbers (from zero to infinity)	Scalar
strings	Strings in Python are an immutable array of characters	Non-Scalar
tuples	Tuples are linear arrays that can store multiple values of any type.	Non-Scalar
dictionaries	A dictionary, stores values and objects within itself indexed by identifiers, or keys. It's an unordered set of items, each of them a key-value pair, where each key is unique.	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.

Although both datatypes are ordered sequences, lists are much more flexible than tuples. This comes from the fact that lists are mutable while tuples are immutable. This allows to modify the different elements inside the list as well as to re-order the elements (sort etc….) which cannot be done with tuples.

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.

This app will need some sort of sequence of words to be studied. For such a sequence we may use either list data type or tuples. We can imagine that the sequence of words may need to be extended by the user over time. Also we may want to give the possibility to sort the element of the list (randomly for instance). Due to the fact that tuples are immutable, tuples don't seem to be the right choice and list datatype would be best in this case to store the words to be studied.

When it comes to each vocabulary word, as each of them includes various properties (definition and categories in particular), we would need to use a dictionary data type where definition would be a string and category would be a string as well.

1.3: Operators & Functions in Python

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!)

```
# Defines travel destinations
travel_destinations = ["Lisbon", "Oslo", "Roma"]

# Asks user to enter a destination
user_destination = str(input("Where do you want to travel?: "))

# Creates a flag to check whether there is a match between the
destination entered by the user and the destinations in the
travel_destinations list
destination_match = False

# Loops through the destination elements available in the
travel_destinations list
for destination in travel_destinations:
  # If the destination entered by the user is equal to one of the
destination available in the travel_destinations list it will print
out "Enjoy your stay in ... !"
  if (user_destination == destination):
    destination_match = True
    print("Enjoy your stay in " + destination + "!")
    # Exit the loop in case there is a match
    break
  else:
    continue

# If at the end of the code after iteration on all the elements of
the travel_destinations list there was no match then the
destination_match flag will remain False and therefore we can print
out "Oops, that destination is not currently available."
if destination_match == False:
  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.

Logical operators in Python include **and**, **or** as well as **not**.

All of these logical operators return a Boolean. Either True or False.

And logical operator returns True in case all statements are True. Otherwise it will return False.

Or logical operator returns True in case at least one statement is True.

Not logical operator returns the opposite of the result (ie. False in case the result is True).

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

Functions in Python are lines of code grouped together and that can be called at some point in the program.
It is useful when we want to use some built-in function (len, sort etc…) for instance and manipulate our code.
It is also helps when we need to repeat some operations several time. It would save some lines of code.
It also make our code much more readable as it allows to group the code and keep it clean.
In addition, it can also prevent the user from modifying some variables in the program (scope).

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.

I believe I could already progress in my general understanding of Python and how it compares to JavaScript in particular (syntax for loops, else if, etc….)

1.4: File Handling in Python

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

As soon as the program is terminated the data will be lost.
In order to make the data persistent, we need to store it somewhere.
Storing the data locally we allow us to re-use it later on. It will not be needed to re-enter it 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?

Pickles are converted complex data packaged into streams of bytes.
Pickles are used to store data. Sometimes we can store simple data into text files. However, for more complex data (such as dictionaries), we would need to store it using pickles instead.

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?

In Python, to find out which directory I am currently in, I would use the **getcwd()** function provided by the **os Module**.

The same module provides also a function that allows to change the current working directory. This function is **chdir()**.

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?

I would use **try-except blocks**. This way instead of terminating the program I could prompt some messages to the user in case there is an 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.

So far, I am impressed by the fact that Python offers great functionalities in terms of files handling as well as error handling in a rather accessible/simple and clear way.

1.5: Object-Oriented Programming in Python

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

Object Oriented Programming is a concept that allows to create objects from classes. This allows to re-use a lot of the code and therefore makes it more concise and more readable.

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

Classes are like blueprints that can be used to create Objects.
For instance, a class Car allows to create several cars objects with parameters such as model, year, milage etc…

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

Method	Description
Inheritance	Inheritence method in OOP gives the possibility to use the properties of the parent class inside a inherited class. This allows in particular to avoid repeating the same code inside several classes.
Polymorphism	In OOP, polymorphism is where a given data attribute or method has the same name across different classes or data types, but performs different operations depending on where it was defined. Therefore even though the two classes have a method with the same name, each method can have its own respective implementation that pertains to its class alone.
Operator Overloading	+, - and other operators such as >, <, >= etc… cannot be used directly inside a Class. If we would use it, we would get a TypeError. To use such operators on a custom class, you need to define our own methods for them. This process is known as operator overloading.

1.6: Connecting to Databases in Python

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

Databases are organized collections of structured information, or data, typically stored electronically in a computer system.
Using databases allows us to make our data persistent and depending on the location of the database (locally or on cloud) data may be accessible to a selection of users.

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

Data type	Definition
INT	Standard integers
VARCHAR(n)	String of variable length, with n representing the maximum number of characters
FLOAT	Floating-point decimal numbers

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

SQLite may be a better choice when Apps only need very simple databases like a simple web page with a limited number of data to store.

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?

Except for the library/package management system, until now I feel that there are not so many differences when programming in Python compared to programming in JavaScript. The approach / logic / concept remains the same in both cases, especially as we try to split our code in several smaller functions so we can re-use it in order not to repeat ourselves.

I can however feel already that Python is more readable and often it may seem unnecessary to add comment in the code as in itself it looks very clear / self-explanatory.

Package management as well as managing the different versions appears however to be much more difficult to handle properly with Python.

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?

For now, I could not see yet the limitations of Python except for the difficulties faced in maintaining a proper environment. This seems to me that it may be difficult to make sure that one developer's environment is strictly the same as the one of another developer.

1.7: Object-Relational Mapping in Python

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

Object Relational Mapper is a tool / a package that allows to interact with our database without using the SQL syntax but instead using Classes and Objects.

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?

App development using Python was very smooth and I appreciated the readability of the code.

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.

. I created a Recipe App using Python
. At first, this App was rather simple and allowed to create recipes and store them locally without the use of any DBMS
. Then I added some functionalities step by step, such as:
    . The possibility to store the recipes on a mySQL database
    . The possibility to calculate the difficulty of the recipes depending on a combination of cooking time and number of ingredients
    . The possibility to search for recipes with a certain number of ingredients
. The different options are accessible through the use of a Menu on the Command Line.

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?

The different concepts around Python Syntax.

b. What's something you're proud of?

I could tackle all the steps rather well and I was not blocked for too long. When this happened, I could solve the issues by myself. Of course, I needed some times to go on Stack overflow but I understand this is inevitable!

c. What was the most challenging aspect of this Achievement?

I believe the most challenging part was the setup of the environment as I encountered some issues with some old Python versions already installed on my computer.

d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?

Definitely. I am very satisfied I could build the code step by step and tackle the issues one after the other. I am rather confident that I could reproduce this process again with some other App.

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

Do not rush, be patient. Take a step at a time.

2.1: Getting Started with Django

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?

Django will help a lot to get started with the project (structure, tools to connect to a database etc…). On the other hand, Django will also come with a very strict structure that the project will have to follow and that will make it much less flexible than a vanilla Python App.

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

MVT architecture does a lot for us in particular when it comes to working with a database. We simply need to specify which items to present to the user, and the framework (Template) will prepare and send it.

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?**

How to implement a full stack project with Django.

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

A clear understanding on how Django is working and be able to do some other Projects.

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

Full stack App development making use of the most appropriate technology / framework depending on the needs.

2.2: Django Project Set Up

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 define the company's website as the Project and then I would break the Project into small Apps. Each App will then be responsible for one specific functionality such as Login, filtering some data, etc…

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

First, I would create a virtual environment then create a project using Django-admin startproject command.

I would then run the first migration in order to create the database (python manage.py migrate) and then I would run the server using python manage.py runserver command.

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

At the beginning, I could use the admin site in order to create my initial data into the database. I could also define some rights for the different users I may want to invite to use the App.

2.3: Django Models

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

A model is a representation of our data and is the single, definitive source of information about your data. It contains the essential fields and behaviors of the data we're storing. Generally, each model maps to a single database table.

One of the key benefits is that once the models are defined, Django manages the communication with the database on our behalf.

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.

It is crucial as it helps design properly our App. In particular, modifying our databases structure / model afterwards can be very much time consuming. If we do the tests first, it helps to define the type of data we need and helps to build the documentation, therefore reducing the need to modify our app afterwards.

2.4: Django Views and Templates

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

Django views are Python functions that takes http requests and returns http response, like HTML documents.

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?

I will use class-based view (CBV). These are easy to reuse and extend compared to the FBVs.

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

Django template language is a simple way provided by Django to **display information from a database or variables**. This can be done using a set of predefined tags.

2.5: Django MVT Revisited

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

Static files are files that are not updated dynamically by the server. (no change once the server is running).

Static files are located in specific parts of the Django project and are managed by Django so the website can give access to these resources to the user.

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	One of the 2 generic class-based views designed to display data. It provides a page representing a list of objects.
DetailView	One of the 2 generic class-based views designed to display data. A base view for displaying a single object.

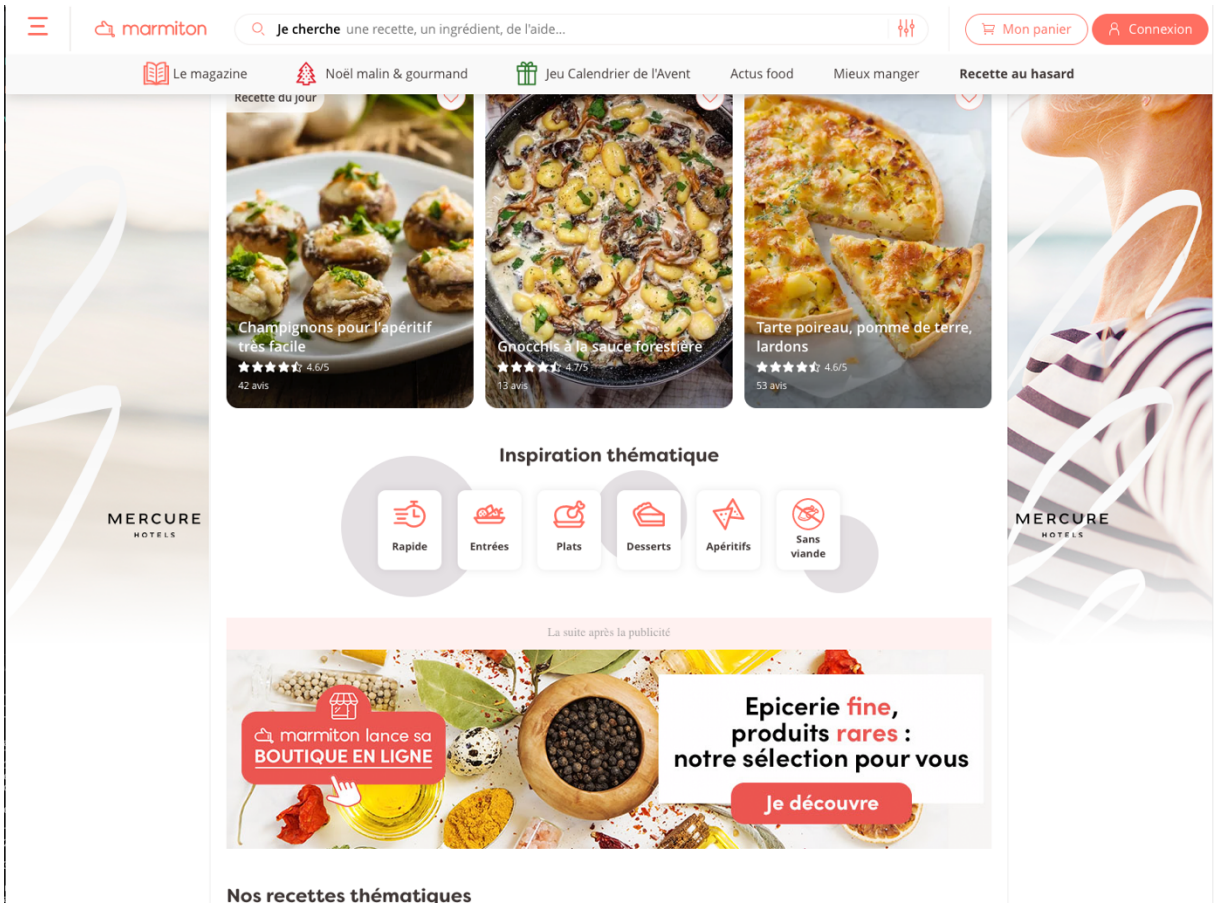
3. You're now more than halfway through Achievement 2! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? You can use these notes to guide your next mentor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

Very much impressed by the tools provided by Django and how efficient it can be to create nice looking website with a database in a limited amount of time. Good to discover the Django way however I feel that a lot of time is needed to check the documentation to make sure that it is used properly.

4. Create a section called "frontend inspirations" in your journal.doc document and add links to your favorite recipe applications (three at most).

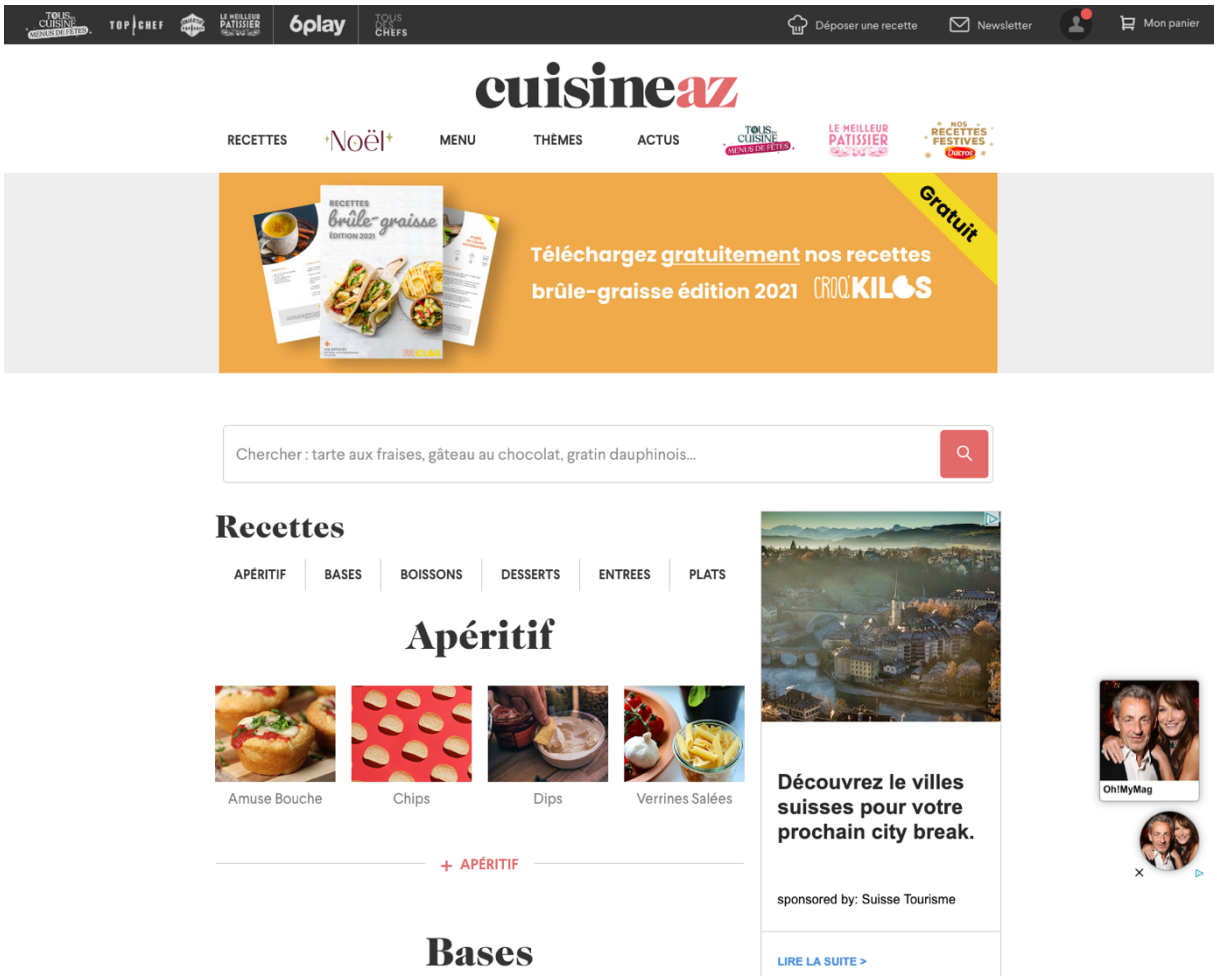
a. **Marmiton**

https://www.marmiton.org/



b. cuisineaz

https://www.cuisineaz.com/categories/



c. **750g**

https://www.750g.com/home_rubrique_-_recettes.htm

Recettes

LES RECETTES PAR CATÉGORIE


Accompagnements


Apéritifs


Bases


Boissons


Boulangerie - Viennoiserie


Confitures


Desserts


Entrées


Gâteaux - Biscuits


Petit-déjeuner


Plats


Pâtisserie

2.6: User Authentication in Django

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.

Sometimes, we may decide to give access to some parts of the website only to some registered users. Sometimes we may want to give some specific rights (like adding some recipes, modifying some recipes etc…) only to some users (for instance the one that have added the specific recipe). Authentication allows exactly to do that.

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

The main steps include:

. Create the view.
. Create the template.
. Specify the URL mapping (although you'll skip this step in this instance).
. Register the URL to the project.

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()	Use authenticate() to verify a set of credentials. It takes credentials as keyword arguments, username and password for the default case, checks them against each authentication backend, and returns a User object if the credentials are valid for a backend. If the credentials aren't valid for any backend or if a backend raises PermissionDenied, it returns None.
redirect()	Returns an HttpResponseRedirect to the appropriate URL for the arguments passed. The arguments could be: A model: the model's get_absolute_url() function will be called. A view name, possibly with arguments: reverse() will be used to reverse-resolve the name. An absolute or relative URL, which will be used as-is for the redirect location. By default issues a temporary redirect; pass permanent=True to issue a permanent redirect.
include()	A function that takes a full Python import path to another URLconf module that should be "included" in this place. Optionally, the application namespace and instance namespace where the entries will be included into can also be specified.

2.7: Data Analysis and Visualization in Django

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.

My favorite application is Blinkist. (https://www.blinkist.com/en/)
Blinkist is a book-summarizing subscription service.

Data that may be collected are:

Number of downloads for each book summary.
Number of summaries selected as favorites by users.
Number of summaries selected as favorites actually read by the user.
Number of summaries shared by the user with other users.
Average number of summaries read per user over a certain period of time (week / month / year).

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

We can evaluate a QuerySet in the following ways:

- Iteration
- Asynchronous iteration
- Slicing
- Pickling/Caching
- repr()
- len()
- list()
- bool()

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.

Dataframe Advantages and Disadvantages:

Advantages	Disadvantages
Less writing	Steep learning curve
Excellent data representation	Difficult syntax
Made for Python	Poor compatibility for 3D matrices
An extensive set of features	Bad documentation

DataFrame can handle more data and are faster than QuerySet.

2.8: Deploying a Django Application

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

First, we create a folder named **static** which we will put inside our **src** folder. We then need to create our CSS (style.css for instance) and JS files (home.js for instance) that will be located inside our newly created static folder.

We then need to refer to the CSS and Javascript files in our HTML templates as follows:

```
<head>
  <link rel="stylesheet" href="static/style.css">
  <script src="static/home.js" defer></script>

</head>
```

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

At first we need to prepare our Django App for Production.
This means that we need to clear sensitive information such as keys, create a .gitignore file to avoid pushing files that will not be needed in our production environment.
We also need to create a Procfile, install Gunicorn, and configure our App to be used on the hosting service provider. After that we have to update our settings.py to handle static files when in production environment and also install whitenoise package. Eventually, we need to create a requirements.txt to replicate our development environment to our production environment

Then we can push our App to GitHub.

Once on GitHub we need to push our App to Heroku or other hosting service provider. The app can then be used and database could be filled out again.

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?

Creating a new app using Python and then Django step by step.

b. What's something you're proud of?

I could solve most of the issues I faced thanks to some internet searches. (Documentation, stackoverflow)

c. What was the most challenging aspect of this Achievement?

Python configuration.

d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

Yes. I was glad to get the chance to build an app using Django and feel confident I could build some other app with it, using the available documentation as well as stackoverflow hints.