Python for Web Developers: Achievement 2 Learning Journal

Pre-work

1. **Was your study routine effective during Achievement 1? If not, what will you do differently during Achievement 2?**

   I tend to be a little loose with my study routine. I try to get an exercise reading and the chapter quiz done each day every couple of days, and if time allows, I'll either get started on taking notes from the reading or doing the task; if time does not allow, I will try to get both done the next day. If I fall behind during the week, I will try to work on the remaining content over the weekend if I don't already have prior obligations or plans. I often have to go to the chiropractor during the week, at least twice weekly, to continue attempting to treat my back injury, so that eats into some of my time from day to day. Additionally, sometimes I'm in so much pain, I can't sit at the desktop to do coding work since that's where I do all of it. This, coupled with the amount of notes I write per chapter, tends to set me back in my course timeline.

2. **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'm proud of my code structure and commenting skills, and I will continue to improve on both of these through Achievement 2 and beyond. I feel like I've also gotten better at working with ChatGPT when using it to assist with coding.

3. **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?**

   My main difficulties during last Achievement revolved around applying previous exercise's content to the next exercise's task coding. Sometimes, it was tricky knowing where to start when writing certain things like try-except blocks or the utilization of certain functions like .join() or .split(). Given more time spent reviewing the chapter readings, my notes, other students' submissions, and working with ChatGPT, I eventually got a better feel for these concepts. If I get stuck with anything in Django, I hope I can reach out to my mentor, other students, or the community online.

Achievement 2: Web Development & Django

Exercise 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?**

   Vanilla Python is better suited for smaller projects with minimal functionality and allows for more control over every aspect of an app, avoiding the constraints of frameworks like Django. However, development can be slower because the main features of the app, like routing and security, have to be built from scratch, rather than being included with a framework.

   Frameworks like Django can be useful for larger, more complex projects as they usually come with built-in features that can handle things like routing, security, functionality, and scalability, cutting down on development time. However, some frameworks, like Django, can have a steep learning curve and be more challenging to work with for those less familiar with them. Additionally, their built-in features can add much additional overhead to the app, resulting in reduced performance. Also, in the case of Django, the app would have to be built the "Django Way", so it would have to follow a specific ruleset to be developed properly.

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

   MVT has a significant advantage over MVC because it relies on an integrated templating system. It simplifies rendering HTML content and allows devs to embed content into their templates.

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.**

   I would like to learn more about what Django can really do on the backend for an app that uses a database, as well as how it renders content on the frontend. I want to add Django as another tool to my skillset that I can use in future projects. After this achievement, I will be working on finishing the Job Prep course and getting ready to go on the hunt.

Exercise 2.2: Django Project Setup

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.**

   Consider the website cloakbrand.com. The website as a whole would be considered a Django project. Just on its home page, which changes depending on what the currently featured apparel collection is, there are various apps, such as dropdown menus for other collections, categories, and information pages. Additionally, the Cloak logo in the center, as well as the search icon and shopping bag icon, would also be considered apps. Further down the page are a large amount of different featured items for purchase, and each one would also be considered an app. Each of these items would also be tied to a database where each of these items from their respective collections would be stored and fetched from to be displayed on the frontend.

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

   To deploy a basic Django app locally, first create a virtual environment and install Django in it. Once the project structure is created, run a migration to create the database of the application. Once everything displays as OK in the terminal, run the server from the terminal, then follow the local host link displayed to view the application in browser to confirm successful deployment.

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

   During development of an application using Django, I would be able to use CRUD operations in order to work with data without having to write additional code. I would also be able to observe any changes to the database and test data in it to check that models and views are working correctly. Additionally, I could manage users of the application and their permissions as well as the content that would be posted or generated by them.

Exercise 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.**

   Django models are used to define the structure of tables in a database and behave similarly to Python classes. They allow you to define a class (model) structure for each app in a Django application, as well as their attributes and methods. Benefits of Django models include its use of ORM, ability to make migrations of changes over time, include validation fields to ensure proper field formatting, and an easy-to-access admin interface.

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's important to write test cases early on in development because they can catch bugs and errors earlier on, saving time down the road. Additionally, they also act as part of the project's documentation. A well-tested project also shows that the code, and its developers, are reliable.

Exercise 2.4: Django Views & Templates

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

   Django views can be set up as functions or classes, and they receive a web request and return a web response. The view of an app is set up in the views.py file of the app directory, where a function is defined to receive the request and return the HttpResponse. Then, in the urls.py file of the app, the defined function is imported, and its URL pattern path is defined. When the server is run and the URL visited, the HttpResponse written in the function will be displayed.

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?**

   Class-based views are preferrable in this case because they are implemented as Python classes and can be reused by using object-oriented programming.

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

   A template is a text file that can be used to generate HTML, XML, CSV, etc. They contain variables, which, when the template is evaluated, get replaced with values. They also contain tags, which control the logic of the template. Variables are enclosed in double curly braces and use a filter to modify their value. Tags provide logic like loops and conditionals.