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.*)

   Let's consider the TryHackMe website. It is a platform for cyber security learning, structured in rooms. The entire website will be called Project in Django terms. Individual components within the site would be apps - login, rooms, leaderboard, blog, student account, learning paths.

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

   Assuming I already have Python installed on the system and the virtual environment extension:
   - Create a folder to store the project on my system
   - In the terminal, create a Python virtual environment
   - Install Django in the virtual environment - pip install django
   - Navigate to the folder that's just been created
   - Create Django project - django-admin startproject <project_name>
   - Navigate to the parent folder of the project just created
   - Deploy project - python manage.py migrate && python manage.py runserver

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

   From my research, here's how I might use the Django admin site during the development of a recipe application:

1. **Data Management**: I can use the Django admin to create, update, delete, and manage recipes, ingredients, categories, and user profiles. This allows me to easily populate the database with test data or real data as the app evolves.

2. **Model Testing**: As I define new models for my app (e.g., Recipe, Ingredient), the admin interface lets me quickly verify that the models and their relationships (such as foreign keys) are working as expected. I can test out different model configurations without needing to create custom forms and views right away.
3. **User and Permissions Control**: The admin site allows me to manage users and set different permissions for them. For example, I can assign certain users as “Recipe Editors” with access to specific parts of the site.
4. **Customization**: During development, I can customize how my models are displayed in the admin (e.g., filtering by recipe category, searching by ingredient), making it easier for me or other team members to manage data efficiently.