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? (Hint: In the Exercise, you saw the example of the CareerFoundry website in the Project and Apps section.)

For a company like **Netflix**, the Django *project* represents the entire website. Each key feature is developed as a Django *app*, for example:

- **accounts** app – for user registration and login
- **movies** app – to list and show movie details
- **watchlist** app – to allow users to save their favorite content
- **subscriptions** app – to handle user plans and payments

Each app is responsible for a specific functionality and they all work together under the main project.

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

- Create a project folder
- Set up and activate a virtual environment using `mkvirtualenv` and `workon`
- Install Django with `pip install django`
- Run `django-admin startproject projectname` (this creates the basic project structure)
- Navigate to the project directory
- Run `python manage.py migrate` to set up the database
- Start the development server with `python manage.py runserver`

- Open the browser and go to `http://127.0.0.1:8000` to see the running project

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

The Django admin site is a built-in feature that allows developers to manage application data through a web interface. It's useful for:

- Creating and managing users
- Adding, editing, and deleting database records
- Testing model interactions
- Managing permissions and user roles

It simplifies backend management without needing to build a separate admin panel.