Exercise 2.6: User Authentication in Django

Learning Goals

- Create authentication for your web application
- Use GET and POST methods
- Password protect your web application's views

Reflection Questions

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.

The importance of incorporating authentication into an application cannot be overstated. It has tons of uses, ranging from being able to control what a customer sees vs. what an employee sees to preserving a users personal data and keeping it shielded from other users. Authentication not only protects data and personal info, but also providers developers with another tool to be able to customize a user's experience.

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

To create a login for a Django web application you need to setup the authentication system by adding 'django.contrib.auth' to the INSTALLED_APPS list in the project's settings.py file. Now using the Django authentication systems User model, you can work on the following steps:

- Create a login view
- Create a logout view
- Map and register the URLS
- Protect views by requiring authentication to load
- Create html templates for views

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()	Takes user credentials as parameters and validates. If user is valid, returns a user object. If not, returns 'none'.
redirect()	Takes URL of page to direct user to, returns view of page and displays corresponding template in browser.
include()	Adds URLs from the apps directory to the main urls.py file in the project directory.