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.

   Incorporating authorization into an application not only helps protect the user’s data, it also allows for a tailored experience for a user to privatize their own user experience.

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

   Django comes with an authentication system that you can add into the project’s settings.py file using ‘django.contrib.auth’ to the INSTALLED_APPS list.
   Then define a user model by creating a new model class in models.py file and set AUTH_USER_MODEL in project’s settings.py file.
   Create a login view using Django’s built-in LoginView and then do the same with the logout view.
   You can then protect the views with authentication using Django’s @login_required decorator or the LoginRequiredMixin mixin.
   Then create templates for the views using the templating system in Django.
   Make sure to include the login and logout URLs in the app’s URLs using the path() function.

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()	Verifies a set of credentials by taking them as keyword arguments and checks them against a authentication backend and returns a user object if credentials are valid
redirect()	Returns an HttpResponseRedirect to the appropriate URL for the arguments passed

include()	Takes a full import path to another URLconf modules that should be “included” in this place.