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.

   Authentication is critical if your app contains views/pages that should only be accessible for certain accounts. Let's take our recipe app as an example:
   As a user I must be the only one with access to my profile page or at least certain functionality like editing user data etc.

2. In your own words, explain the steps you should take to create a login for your Django web application.
   A login needs to have a view, a template and a registered URL, basically like any other app.
   The view needs a form for entering user credentials and needs to use the 'authenticate' method imported via
   `from django.contrib.auth import authenticate`.

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()	Use `authenticate()` to verify a set of credentials. It takes credentials as keyword arguments, `username` and `password` for the default case, checks them against each authentication backend, and returns a `User`object if the credentials are valid for a backend. If the credentials aren't valid for any backend or if a backend raises `PermissionDenied`, it returns `None`. Source: https://docs.djangoproject.com/en/5.1/topics/auth/default/#django.contrib.auth.authenticate
redirect()	To make your life easier, Django provides the versatile shortcut function you've already seen in the introduction: `django.shortcuts.redirect()`. You can call this function with: • A model instance, or any other object, with a `get_absolute_url()` method • A URL or view name and positional and/or keyword arguments • A URL