

Python for Web Developers Learning Journal

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 authentication into an application is vital because it ensures that only authorized users have access to specific features and content which protects their personal data and information.

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

In order to create a login for a Django application, you must first write two import statements in the views.py file of the project folder, including “from django.contrib.auth import authenticate, login” and “from django.contrib.auth.forms import AuthenticationForm.” From there, you will create the view and template, and then register the URL to the project. Afterwards, you will add a single-click login option onto the homepage to direct users to the authentication form. Finally, you will protect the views you only want to display after authentication.

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()	Authenticate() is used to verify a set of credentials. It takes credentials as keyword arguments and checks them against the authentication backend and will return a user if valid and none if not.
redirect()	Redirect() returns an HttpResponseRedirect to the appropriate URL for the arguments passed.
include()	Include() imports and renders a template, allowing inclusion of other templates within a template.

