

Exercise 2.6: User Authentication in Django

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 a critical component of any application that deals with sensitive data or user accounts. It verifies the identity of users before granting them access to the application's resources, ensuring that only authorized individuals can perform specific actions.

Let's consider an online banking application as an example. Without authentication, anyone could access the application and view sensitive account information or transfer funds without authorization. By implementing authentication, the application can verify the identity of the user before allowing them to perform any actions.

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

First, you need to create a view for the login functionality of your app. The view should first check that when the user hits the 'login' button, a 'POST' request with required data is sent using `AuthenticationForm()`, a Django built-in function, then it should check if the user is valid - during which it sends an error message if it's not — using another Django built-in function, `authenticate()`.

Next, a successful user is logged in using yet another Django built-in function `login()` and taken/redirected to the requested page.

In the next step, you create a template for the login form and register the URL for the new login view you created.

Finally, you protect the views you want only authenticated users to access bypassing `LoginRequiredMixin` as a parameter in class-based views and on function-based views add `@login_required` decorator before the definition.

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
<code>authenticate()</code>	is a django function which takes user credentials as parameter and checks if user is valid against the backend data. If user is valid it returns a user object if user is not valid it returns None.
<code>redirect()</code>	is a function from Django module <code>django.shortcuts</code> that takes URL of the page you want a user to be directed to and it returns the view of that page which in turn the view displays the corresponding template.
<code>include()</code>	is Django function used add urls from apps directory to the main <code>urls.py</code> file in the project directory.