

## Exercise 2.6: User Authentication in Django

### Reflection Questions :

- *In your own words, write down the importance of incorporating authentication into an application. You can take an example application to explain your answer.*
  - **A:** Authentication is important to establish user identification, personalization of users' activity and saved information, data security such as personal and financial information, access control to define where various users can and can not navigate within the app, and establishing user comfort and safety; just to name a few.
- *In your own words, explain the steps you should take to create a login for your Django web application.*
  - **A:**
    - Create a Login View – This is recorded for Django to find in the 'views.py'. This will house the logic for the app in routing the user upon successful login as well as the app's behavior in response to a failed login.
    - Create Login Templates – This is the '.html' document that houses the interface code to render the input fields that the user will use to enter their credentials.
    - Define URLs – The URL endpoints need to be defined in the 'urls.py' to establish where the user get routed via URL endpoints.
    - Create Logout View – In kind with the Login View, this houses the logic, in the 'views.py', for the app's behavior when the users' initiate a logout.
    - Assign Protected Views – Designate which pages with the '@login\_required' decorator that Django uses in the 'views.py' to check that the user is authorized by a successful login to access that page.

- *Look up the following three Django functions on Django's official documentation and/or other trusted sources and write a brief description of each.*

- **A:**

Function	Description
authenticate()	Checks a set of credentials against the user model in the database, returning the user object if the credentials are valid.
redirect()	Shortcut to create HTTP redirect responses, by taking a URL as an argument to redirect the user to the specified URL.
include()	Allows the developer to reference other URL configurations from different apps or files in their URL patterns, making organization and modularization of URL routing easier.