# 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 important if we want to provide (or withhold) certain features and/or information according to user rights. For example, in my recipe app only authenticated users can see the recipe details.

2. **In your own words, explain the steps you should take to create a login for your Django web application.**
   To create a login for my Django web application I should follow these four steps:
   1. Create the view
   2. Create the template
   3. Specify the URL mapping
   4. Register the URL to the project

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()`** verifies a set of credentials, username and password by default, 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`. |

| | |
|---|---|
| redirect() | Returns an `HttpResponseRedirect` to the appropriate URL for the arguments passed.<br><br>The arguments could be:<br><br>• A model: the model's `get_absolute_url()` function will be called.<br>• A view name, possibly with arguments: `reverse()` will be used to reverse-resolve the name.<br>• An absolute or relative URL, which will be used as-is for the redirect location.<br><br>By default issues a temporary redirect; pass permanent=True to issue a permanent redirect. |
| include() | A function that takes a full Python import path to another `URLconf` module that should be "included" in this place. Optionally, the application namespace and instance namespace where the entries will be included into can also be specified.<br>Usually, the application namespace should be specified by the included module. If an application namespace is set, the namespace argument can be used to set a different instance namespace.<br>`include()` also accepts as an argument either an iterable that returns URL patterns or a 2-tuple containing such iterable plus the names of the application namespaces.<br><br>Parameters:<br>• `module` – `URLconf` module (or module name)<br>• `namespace` (str) – Instance namespace for the URL entries being included<br>• `pattern_list` – Iterable of path() and/or re_path() instances.<br>• `app_namespace` (str) – Application namespace for the URL entries being included |