Reflection Questions Exercise 2.6

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.**
   An obvious reason to include authentication is security. Something like a shopping web application that handles users' financial information would benefit from adding that layer of protection for their customers. Another reason is personalization, like what Career Foundry does with their website. Once a user logs it, the dashboard displayed is automatically customized. This builds an app's versatility and overall appeal to the user.
2. **In your own words, explain the steps you should take to create a login for your Django web application.**
   I would start by putting together the view (on the project level this time), which would generate a `POST` request when the 'login' button is clicked and bind that request to a variable. The form submission would have to be validated using Django's built in functions and if authenticated, the user would be re-directed to the app home page. Once the view is put together, I'd move on to the HTML template and map that URL in the `urls.py` file. That would complete the login functionality, so it would be best to follow that up with a quick logout FBV that re-directs the user to the login page.
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.**
   `authenticate()` - function that takes each login credential as a keyword argument and checks them against the authentication backend. It returns User objects if they're valid, and if not will return `“none”`
   `redirect()` - takes either a view or another object, or a hardcoded URL as an argument. That argument is used to figure out a re-direct URL. This function is often used when implementing login functionality – once a user is authenticated, they are *redirected* to the home page. Once they log out, they are redirected back to the login page.
   `include()` – a function used for mapping URLs from different files together. It takes a Python import path as an argument (to a URL conf module) and will user URL paths it finds in the file to map the desired URL.