Django Login System

- Django has a registration app for login/logout, sign up, and password registration
- Add Django.contrib.auth in INSTALLED_APPS to use the login system and import it into views.py
- Login() takes an HttpRequest and User object in and saves the user's ID in the session using Django's session framework
- By default the login page will render the registration/login.html template. It uses a simple html form with POST to pass the data between pages in the session.
- The login template name is passed as a parameter in urls.py if the default page is not being used. Don't use the default page unless the site is super basic and ugly.
- In settings.py, set up which URL to redirect to after successful authentication and login of a user.
- Django also includes a simple permissions system to create admins, regular users, site owners, etc. with different permissions to access different pages and therefore carry out different tasks.
- Permissions include has_add_permission(), has_change_permission(), and has_delete_permission()
- User objects have groups and user_permissions fields, both of which are many-to-many fields.
- Groups can be created in admin and permissions can be assigned to users or to groups

- Example of user login and authentication:

```python
from django.contrib.auth import authenticate, login

def my_view(request):
    username = request.POST['username']
    password = request.POST['password']
    user = authenticate(username=username, password=password)
    if user is not None:
        login(request, user)
        # Redirect to a success page.
        ...
    else:
        # Return an 'invalid login' error message.
        ...
```

- Password reset provides a form for emailing the user a password resent link, a view confirming the link was sent, a form for entering a new password and a view that informs the user that the password has been changed successfully.