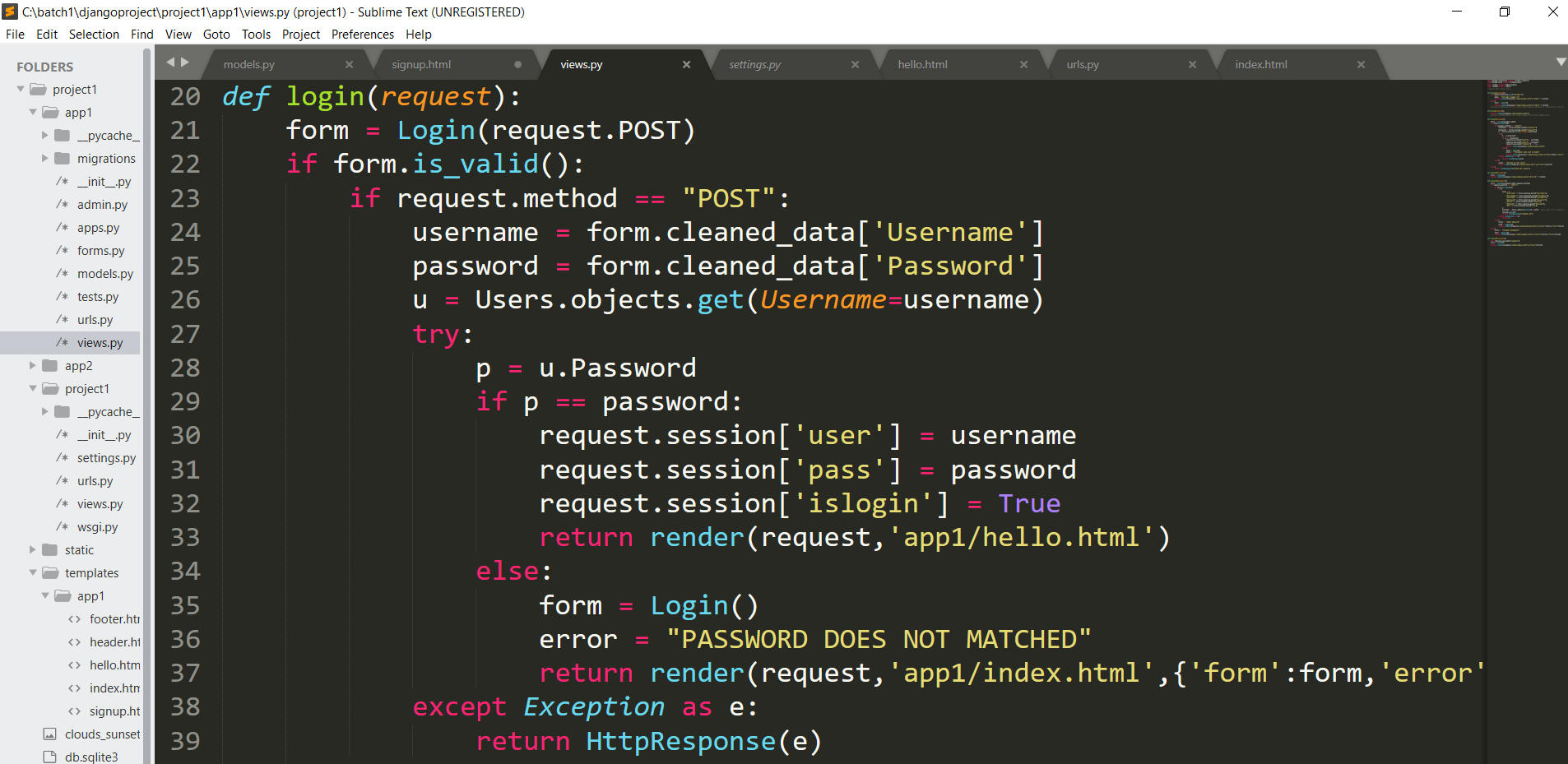
# Sessions

Django provides full support for anonymous sessions. The session framework lets you store and retrieve arbitrary data on a per-site-visitor basis. It stores data on the server side and abstracts the sending and receiving of cookies.

Sessions are implemented via a piece of [middleware](https://docs.djangoproject.com/en/2.1/ref/middleware/).

To enable session functionality, do the following:

* Edit the [**MIDDLEWARE**](https://docs.djangoproject.com/en/2.1/ref/settings/#std:setting-MIDDLEWARE) setting and make sure it contains **'django.contrib.sessions.middleware.SessionMiddleware'**. The default **settings.py** created by **django-admin startproject** has **SessionMiddleware** activated.
* When **SessionMiddleware** is activated, each **[HttpRequest](https://docs.djangoproject.com/en/2.1/ref/request-response/" \l "django.http.HttpRequest" \o "django.http.HttpRequest)** object – the first argument to any Django view function – will have a **session** attribute, which is a dictionary-like object.
* You can read it and write to **request.session** at any point in your view



Here ‘user’ is an key and ‘username’ on right side is its value.

* To get the value of the session[‘key’] following syntax is used.



You can use the value of session whenever you have to render a page when the user is logged in or logged out.

* To delete the session[‘key’] :

