Installation guide:
Prerequisites

1. Git is installed on your computer, You can install it from this link https://git-scm.com/downloads
2. Python is installed. You can download the most recent version of python from this link https://www.python.org/

Step One:

The first step to installing this project is to first clone the repository. This can be easily done using windows command prompt. You first need to navigate to the folder where you want to store the project, then run this command

```
git clone https://github.com/NateClarkTech/Software-Engineering-3
```

After that, you are going to want to Open up a python terminal and begin installing the dependencies.

First and foremost is Django, which you can install with this command

```
Pip install django
```

The next dependency is Bleach,

```
Pip install bleach
```

And run the following commands

```
Pip install python-dotenv
Pip install requests
Pip install pillow
```

After these are all installed, the majority of the app will be able to run on your local machine. However next we must setup the Spotify API.
Resources for setting up the spotify API can be found here
https://developer.spotify.com/documentation/web-api

What you are going to want to setup is a spotify developer account, And create a new app, Name it as you wish.

Once you have an account, and an app, you need to get a client_id and a client_secret.
CLIENT_ID: This will be the unique ID assigned to your app
CLIENT_SECRET: This will be the unique key that will be used to authorize your api calls.

Once you have these, make sure to keep them private, and do not let them be uploaded to github.

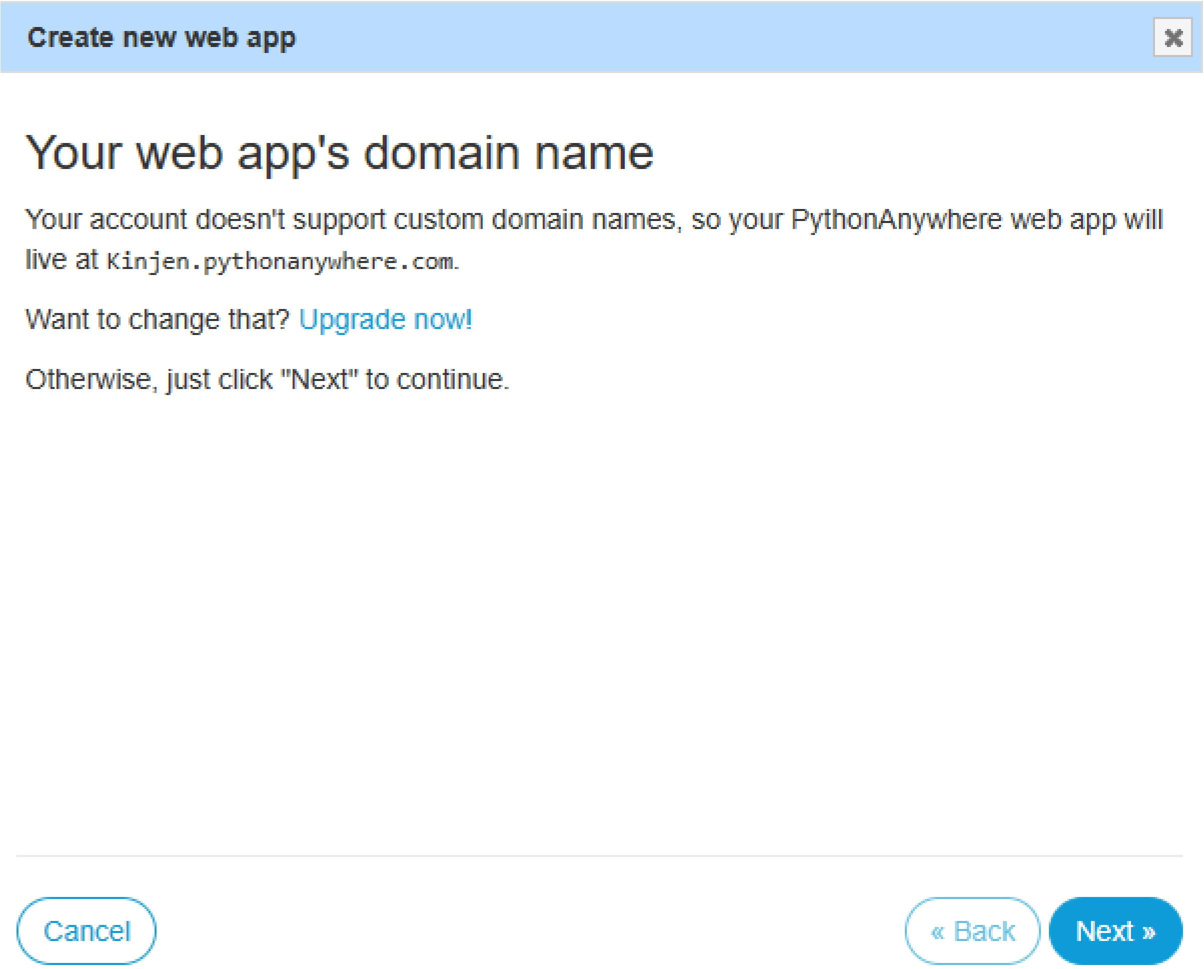
With both of these, You then need to create a .env file inside the IdeaBoards folder
Name it exactly .env
Inside that app you will write this (replace the parentheses with your client_id and secret)

```
CLIENT_ID="(Add your client id here)"
CLIENT_SECRET="(Add your client secret here)"
```

After this is functioning, everything should be ok to run the app fully locally. To deploy this app, you simply need to follow the instructions below.

Deployment guide:

In order to deploy this app on to the internet, We can first begin by creating an account on pythonanywhere
https://www.pythonanywhere.com/
From here, you will need to navigate to the web app tab from the dashboard. And create a new web app



Create new web app

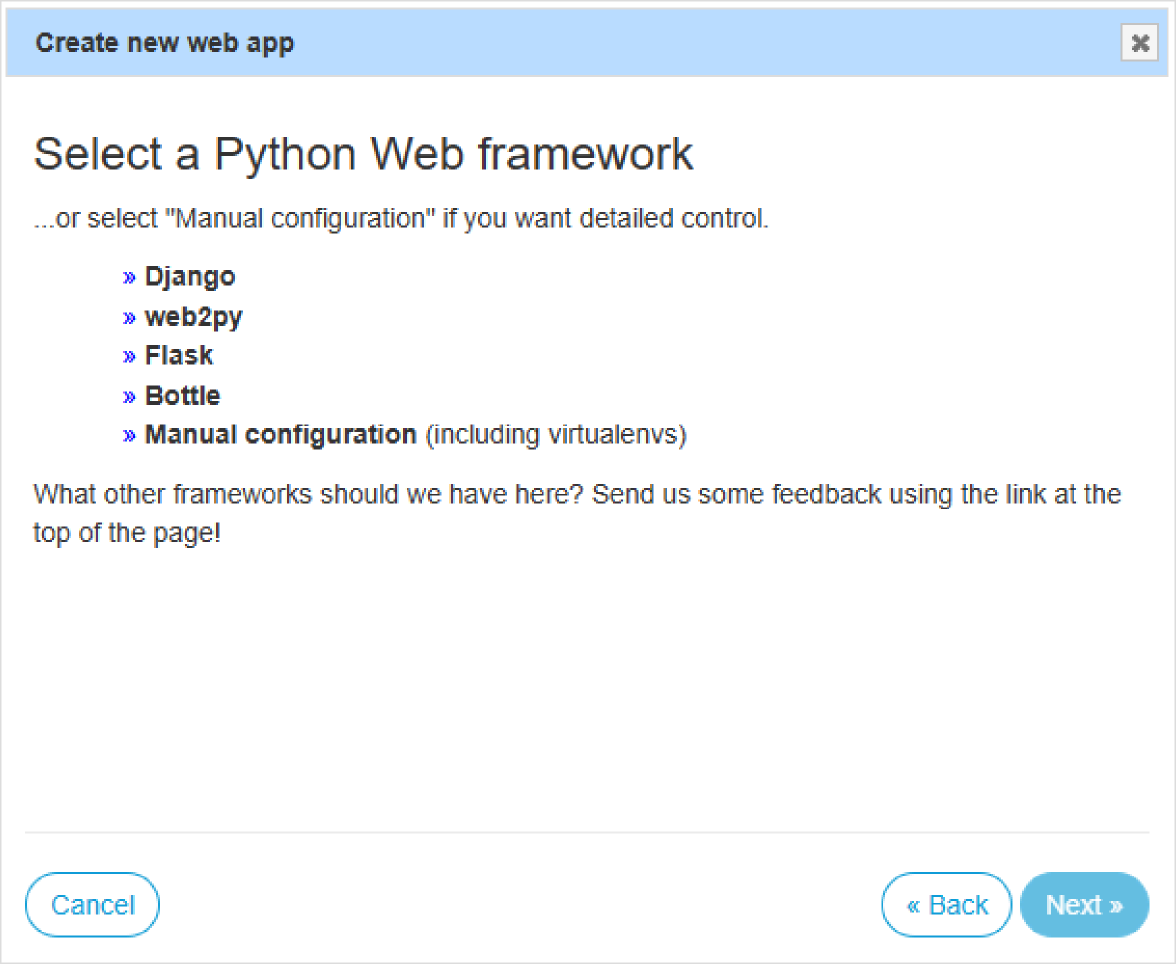
Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `Kinjen.pythonanywhere.com`.

Want to change that? Upgrade now!

Otherwise, just click "Next" to continue.

Selecting next, we will then need to select manual configuration



Create new web app

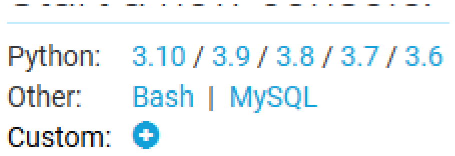
Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » **Django**
- » **web2py**
- » **Flask**
- » **Bottle**
- » **Manual configuration** (including virtualenvs)

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

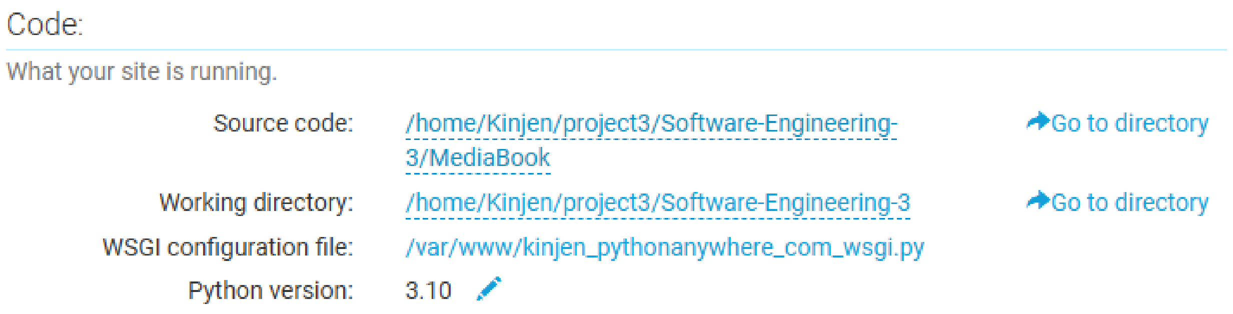
And select PYTHON 3.10
From there, you can select next, and you will be taken to your new apps web app dashboard.
After that, you will need to head over to the consoles tab and create a new bash console



Python: 3.10 / 3.9 / 3.8 / 3.7 / 3.6
Other: Bash | MySQL
Custom:

Once the console is open, you will need to run this command
Git clone https://github.com/NateClarkTech/Software-Engineering-3

This will download the application onto the server
Next, we will need to setup the project

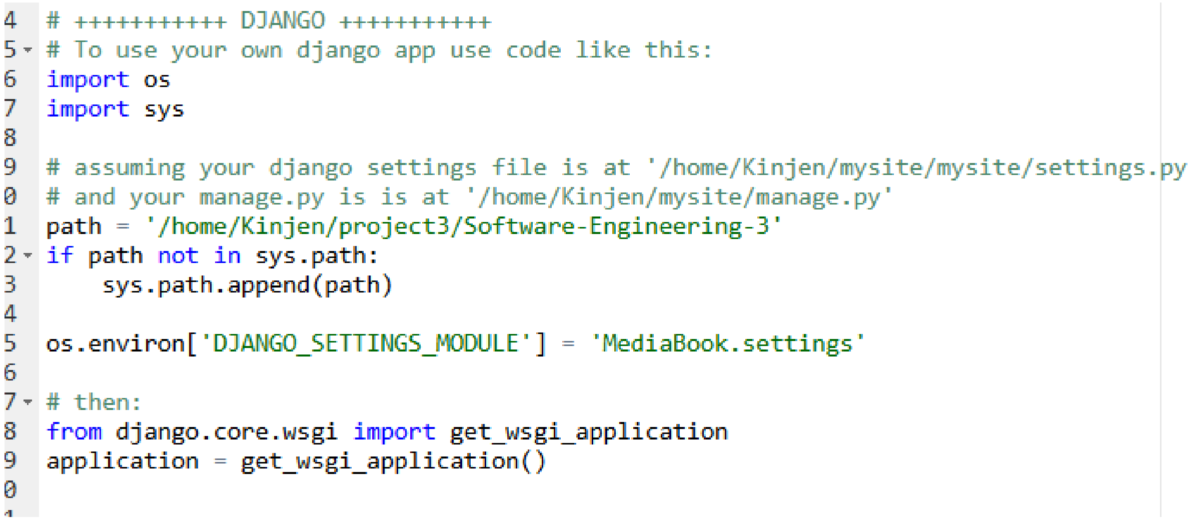


Code:

What your site is running.

Source code:	/home/Kinjen/project3/Software-Engineering-3/MediaBook	Go to directory
Working directory:	/home/Kinjen/project3/Software-Engineering-3	Go to directory
WSGI configuration file:	/var/www/kinjen_pythonanywhere_com_wsgi.py	
Python version:	3.10	

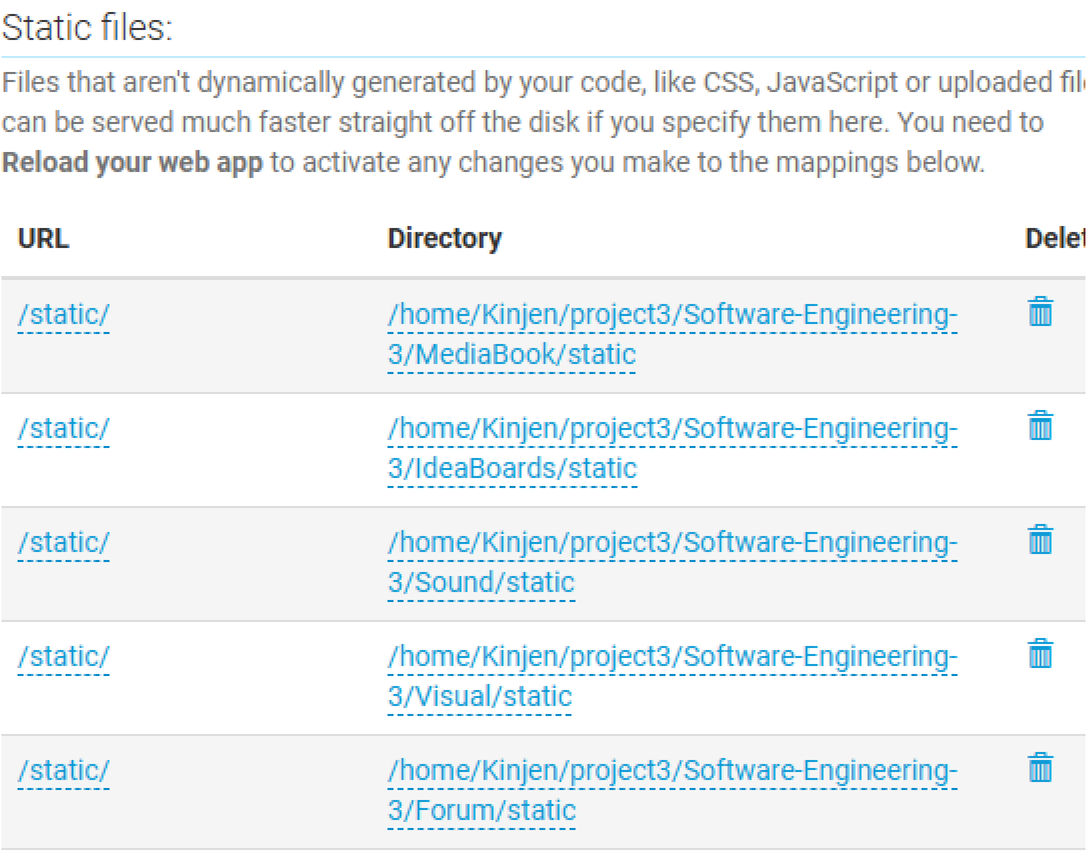
Under the code section, you will need to tell the application where to find the files for the project. In the above image, you can see that we need to find the path to the MediaBook Folder for the source code, and the working directory will be in the Software engineering 3 directory. Next, we are going to open the WSGI configuration file by clicking on the link. And navigate to the django section



```
4  # +++++++++++ DJANGO +++++++++++
5  # To use your own django app use code like this:
6  import os
7  import sys
8
9  # assuming your django settings file is at '/home/Kinjen/mysite/mysite/settings.py'
0  # and your manage.py is is at '/home/Kinjen/mysite/manage.py'
1  path = '/home/Kinjen/project3/Software-Engineering-3'
2  if path not in sys.path:
3      sys.path.append(path)
4
5  os.environ['DJANGO_SETTINGS_MODULE'] = 'MediaBook.settings'
6
7  # then:
8  from django.core.wsgi import get_wsgi_application
9  application = get_wsgi_application()
0
```

From here, you are going to want to uncomment the lines you see in the image, and change the path to more resemble what is shown here. You want to direct it to the folder that holds the manage.py file. Next you are going to replace line 5 yourapp.settings with MediaBook.settings. Click save and you should be good to go. (note, replace Kinjen with your username, and you may need to remove project 3 from the path.

Next you need to tell the app where all of the static files are.



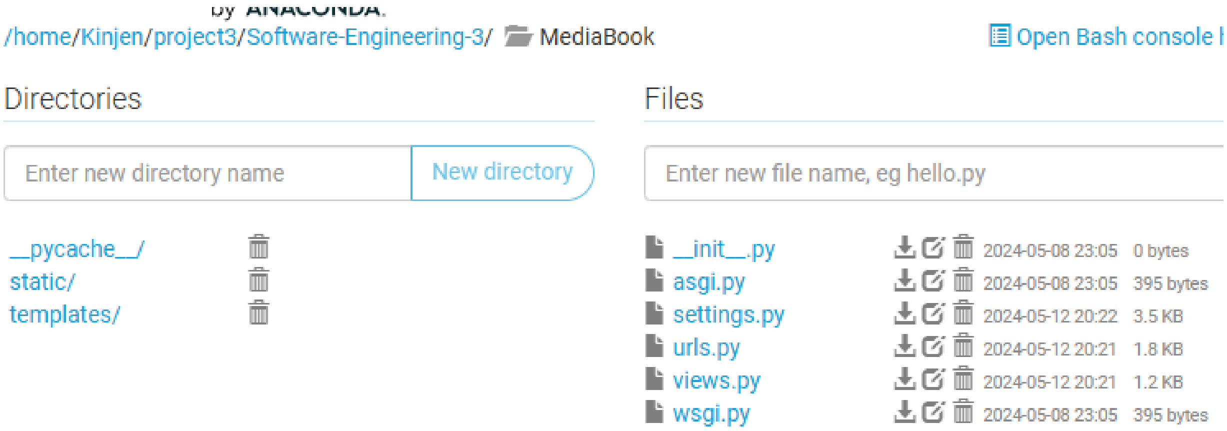
Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded fil
can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delet
/static/	/home/Kinjen/project3/Software-Engineering-3/MediaBook/static	🗑
/static/	/home/Kinjen/project3/Software-Engineering-3/IdeaBoards/static	🗑
/static/	/home/Kinjen/project3/Software-Engineering-3/Sound/static	🗑
/static/	/home/Kinjen/project3/Software-Engineering-3/Visual/static	🗑
/static/	/home/Kinjen/project3/Software-Engineering-3/Forum/static	🗑

You simply need to write the path of each static folder into this section.

Finally, you will need to add your app to the “allowed host” section of the settings file.



by ANACONDA.

/home/Kinjen/project3/Software-Engineering-3/ MediaBook

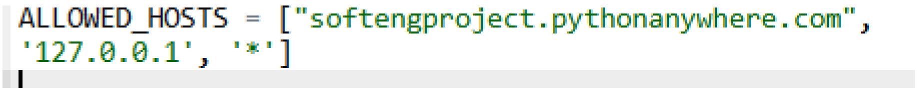
Directories

__pycache__/
static/
templates/

Files

__init__.py	2024-05-08 23:05	0 bytes
asgi.py	2024-05-08 23:05	395 bytes
settings.py	2024-05-12 20:22	3.5 KB
urls.py	2024-05-12 20:21	1.8 KB
views.py	2024-05-12 20:21	1.2 KB
wsgi.py	2024-05-08 23:05	395 bytes

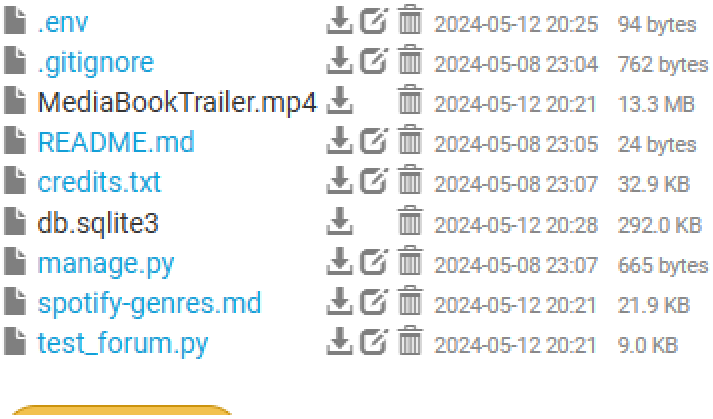
Open the settings file, and replace softengproject.pythonanywhere.com with your app url.



```
ALLOWED_HOSTS = ["softengproject.pythonanywhere.com",
'127.0.0.1', '*']
```

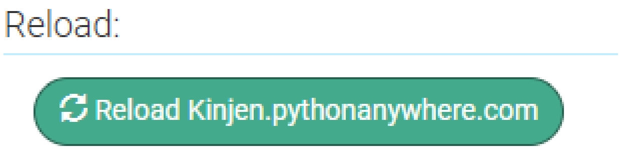
Next you are going to need to setup the Spotify API. Much like installing it on the local host, You will need to setup the spotify application, and get a client_id and client_secret

Creating an env file. If you have followed the local host installation guide, you simply need to upload the .end file you created up there to the Software_engineering_3 folder.



.env	2024-05-12 20:25	94 bytes
.gitignore	2024-05-08 23:04	762 bytes
MediaBookTrailer.mp4	2024-05-12 20:21	13.3 MB
README.md	2024-05-08 23:05	24 bytes
credits.txt	2024-05-08 23:07	32.9 KB
db.sqlite3	2024-05-12 20:28	292.0 KB
manage.py	2024-05-08 23:07	665 bytes
spotify-genres.md	2024-05-12 20:21	21.9 KB
test_forum.py	2024-05-12 20:21	9.0 KB

And just click the reload app button on the web dashboard.



From here, everything should be functioning.

Including env files in deployment:
Back inside your project directory in the bash console, we need to create a .env file at the same level as the manage.py file.

```
touch .env
nano .env
```

Inside of the terminal editor, add your secret key like below. Make sure to save the file.

```
export SECRET_KEY="mysupersupersecretkey"
```

Next, we need to install dotenv so we can use our environment variable file.

```
pip install python-dotenv
echo python-dotenv >> requirements.txt
```

This installs dotenv and adds it to our requirements.txt file.

How to redeploy changes made to the application: