[Deploying an existing Django project on PythonAnywhere](https://help.pythonanywhere.com/pages/DeployExistingDjangoProject/)

You Need to Deploy your Model at Github

And copy the link for clone process as follow

A screenshot of a computer

Description automatically generated

Firstly sign up for Pythonanywhere

A screenshot of a computer

Description automatically generated

Create the Account :- Which we would use as account to host the project of Django,Flask,etc

A screenshot of a computer

Description automatically generated

Now lets start Uploading the project:-

Go to the $Bash

A black rectangular object with white border

Description automatically generated

## **Uploading your code to PythonAnywhere**

**Bash Console**: $ git clone https://github.com/myusername/myproject.git

## **Create a virtualenv and install Django and any other requirements**

$ mkvirtualenv --python=/usr/bin/python3.10 mysite-virtualenv

(mysite-virtualenv)$ pip install django

*# or, if you have a requirements.txt:*

(mysite-virtualenv)$ pip install -r requirements.txt

**Warning**: Django may take a long time to install. PythonAnywhere has very fast internet, but the filesystem access can be slow, and Django creates a lot of small files during its installation. Thankfully you only have to do it once!

**TIP**: if you see an error saying *mkvirtualenv: command not found*, check out [*InstallingVirtualenvWrapper*](https://help.pythonanywhere.com/pages/InstallingVirtualenvWrapper).

For example :- see this -**A screen shot of a computer program

Description automatically generated**

**Setting up your Web app and WSGI file**

**Go to dashboard and Click Web**

1. The path to your Django project's top folder -- the folder that contains "manage.py", eg */home/myusername/mysite*
2. The name of your project (that's the name of the folder that contains your settings.py), eg *mysite*
3. The name of your virtualenv, eg *mysite-virtualenv*

#### **Create a Web app with Manual Config**

Head over to the **Web tab** and create a new web app, choosing the "Manual Configuration" option and the right version of Python (the same one you used to create your virtualenv).

**A screenshot of a web framework

Description automatically generated**

**NOTE:** Make sure you choose **Manual Configuration**, not the "Django" option, that's for new projects only

A screenshot of a computer

Description automatically generated

#### **Enter your virtualenv name**[**¶**](https://help.pythonanywhere.com/pages/DeployExistingDjangoProject/#enter-your-virtualenv-name)

**A screenshot of a message

Description automatically generated**

You can just use its short name "mysite-virtualenv", and it will automatically complete to its full path in /home/username/.virtualenvs.

#### **Optional: enter path to your code**

Although this isn't necessary for the app to work, you can optionally set your working directory and give yourself a convenient hyperlink to your source files from the web tab.

Enter the path to your project folder in the Code section on the web tab, eg /home/myusername/mysite in **Source code** and **Working directory**

**A close-up of a message

Description automatically generated**

#### **Edit your WSGI file**

One thing that's important here: your Django project (if you're using a recent version of Django) will have a file inside it called wsgi.py. This is not the one you need to change to set things up on PythonAnywhere -- the system here ignores that file.

Instead, the WSGI file to change is the one that has a link inside the "Code" section of the **Web tab** -- it will have a name something like /var/www/yourusername\_pythonanywhere\_com\_wsgi.py or /var/www/www\_yourdomain\_com\_wsgi.py.

Click on the WSGI file link, and it will take you to an editor where you can change it.

Delete everything except the Django section and then uncomment that section. Your WSGI file should look something like this:

You can get the location of your project in bash by using pwd command

*# +++++++++++ DJANGO +++++++++++*

*# To use your own Django app use code like this:*

**import** **os**

**import** **sys**

*# assuming your Django settings file is at '/home/myusername/mysite/mysite/settings.py'*

path = '/home/myusername/project'

**if** path **not** **in** sys.path:

sys.path.insert(0, path)

os.environ['DJANGO\_SETTINGS\_MODULE'] = 'app.settings'

*## Uncomment the lines below depending on your Django version*

*###### then, for Django >=1.5:*

**from** **django.core.wsgi** **import** get\_wsgi\_application

application = get\_wsgi\_application()

*###### or, for older Django <=1.4*

*#import django.core.handlers.wsgi*

*#application = django.core.handlers.wsgi.WSGIHandler()*

* Be sure to substitute the correct path to your project, the folder that contains manage.py, which you noted above.
* Don't forget to substitute in your own username too!
* Also make sure you put the correct value for DJANGO\_SETTINGS\_MODULE.
* This guide assumes you're using a recent version of Django, so leave the old wsgi.WSGIHandler() code commented out, or better still, delete it.

Save the file, then go and hit the **Reload** button for your domain. (You'll find one at the top right of the wsgi file editor, or you can go back to the main web tab)

Open the setting.py file in our project and search Allowed host[]:-

Allowed host =[‘your-username-project’]

this you can get at the web tab in the pythonanywhere

Check the static file location in the bash and replace the location in the settings.py

STATIC\_ROOT = "/home/myusername/myproject/static"

*# or, eg,*

STATIC\_ROOT = os.path.join(BASE\_DIR, "static")

Check the same for the templetes location in the bash and replace the location in the settings.py

### **Set up a static files mapping**

Finally, set up a static files mapping to get our web servers to serve out your static files for you.

* Go to the **Web** tab on the PythonAnywhere dashboard
* Go to the **Static Files** section
* Enter the same URL as STATIC\_URL in the **url** section (typically, /static/)
* Enter the path from STATIC\_ROOT into the **path** section (the full path, including /home/username/etc)

Then hit **Reload** and test your static file mapping by going to retrieve a known static file.

Eg, if you have a file at /home/myusername/myproject/static/css/base.css, go visit [*http://www.your-domain.com/static/css/base.css*](http://www.your-domain.com/static/css/base.css)

You can do those this with the Media also respective

## **Database setup**

If, like most sites, your site uses a database, you'll need to set that up. Go to the **Consoles tab**, start a bash console, use cd to navigate to the directory where your Django project's manage.py lives, then run

Python manage.py migrate

**You can go and Reload the file at Web and Click on the User-name\_Project.com**

**A screenshot of a computer

Description automatically generated**

**If you face error you can delet those files and console and do the steps from beginning to end**

**-Thankyou**