Django is one of the best python frameworks used for developing web applications. After successful development and testing a Django web application offline on your local machine, it is always good to make it available to users. This can be done by uploading your site on various web hosting companies or by using pythonanywhere to test your site online and check its performance. I had a hard time uploading my first Django application to pythonanywhere to a point where I almost gave up. After some research, I was able to successfully clone my Django application from Github and upload it to Pythonanywhere. I have composed a series of steps and configurations to guide you until your site becomes live. Let’s get started!

Django is one of the best python frameworks used for developing web applications. We often develop web applications and run it locally. However, many developers are keen to deploy their projects or applications on live server.

There are multifarious services provider who provide host to deploy project. But,in this blog we are going to deploy web application on **PythonAnyWhere** development environment to deploy our web application.

We would follow the following steps to easily deploy our project with MYSQL database on **PythonAnyWhere**.

**Local Server:**

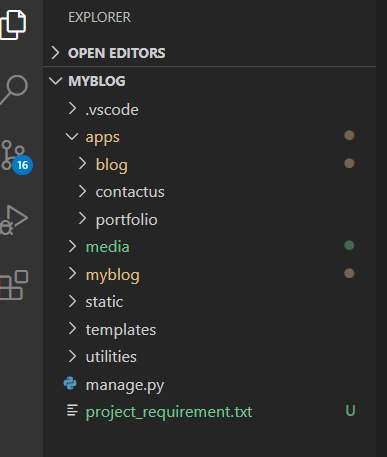
* First run your project at your local machine and check everything is running.
* Generate project\_requirement.text file from your local machine.
* Commit and push your code to Github account **Or** just zip your project.

We are done at our local machine , Now lets jump into pythonanywhere.

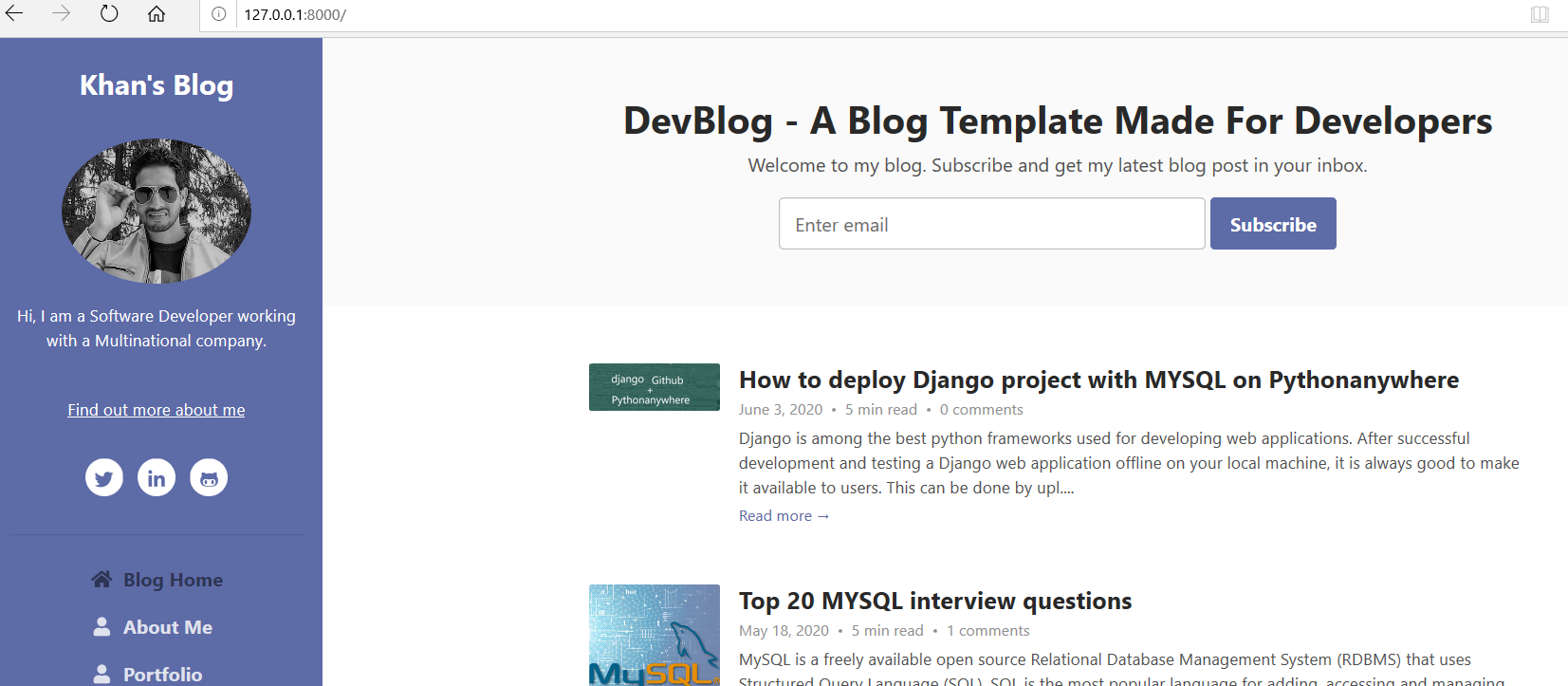
**PythonAnyWhere**:

* Create your account.
* Database Configurations.
* Project configurations.
* Refresh web server and all done

Since we are going to deploy our project to live host, therefore, it is essential to run and test our project on local environment/machine before making it live. Here is my folder structure and running app on local machine.



Locally running blog:



If all thing is running fine just generate project\_requirement.txt file. This file is very important file which contain all installed modules/packages that are required in server machine in order run project smoothly.

In order to generate project\_requirement.txt , type in project command line interface:

**>> Pip freeze > project\_requirement.txt**

**3.> Push code to Githu or zip it**

Run following git command to push your code into Github.

>> git status # To check the status of project

>> git add . # add all updated file if any

>> git commit -m “commit complete code”

>> git push -u origin master

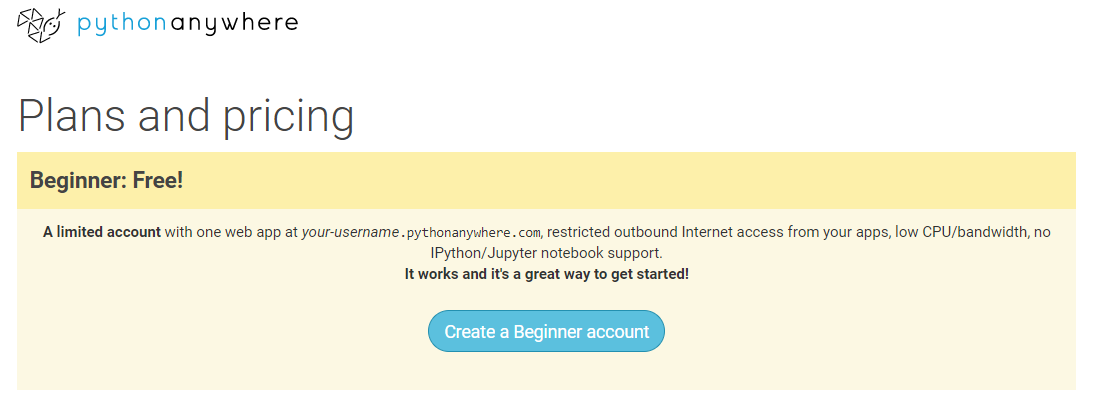
In case your are not using git, just Zip your main project.

Now it all done at local side.

Python anywhere:

First create your account from this link <https://www.pythonanywhere.com/pricing/>

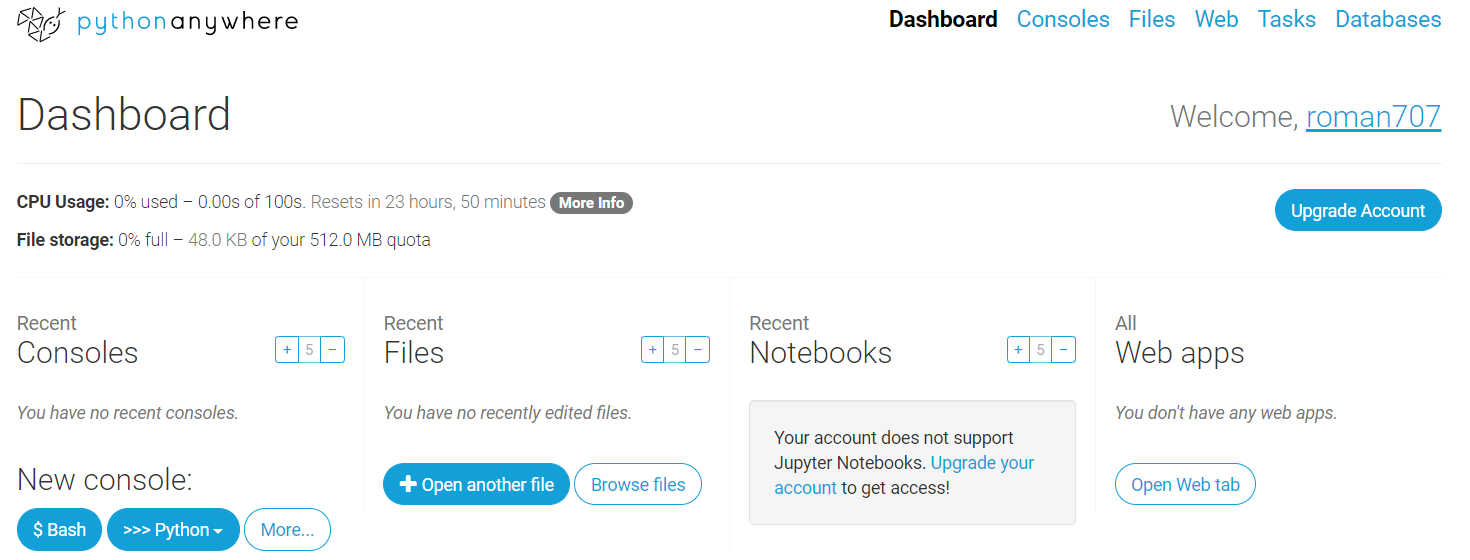
Click on this linke  https://www.pythonanywhere.com/pricing/



In this blog we are creating free account so that we can test our project.

So create your account and confirm your email address otherwise you would not be able to deploy your project.

After creating and confirming your email address successfully just come to Dashbord.



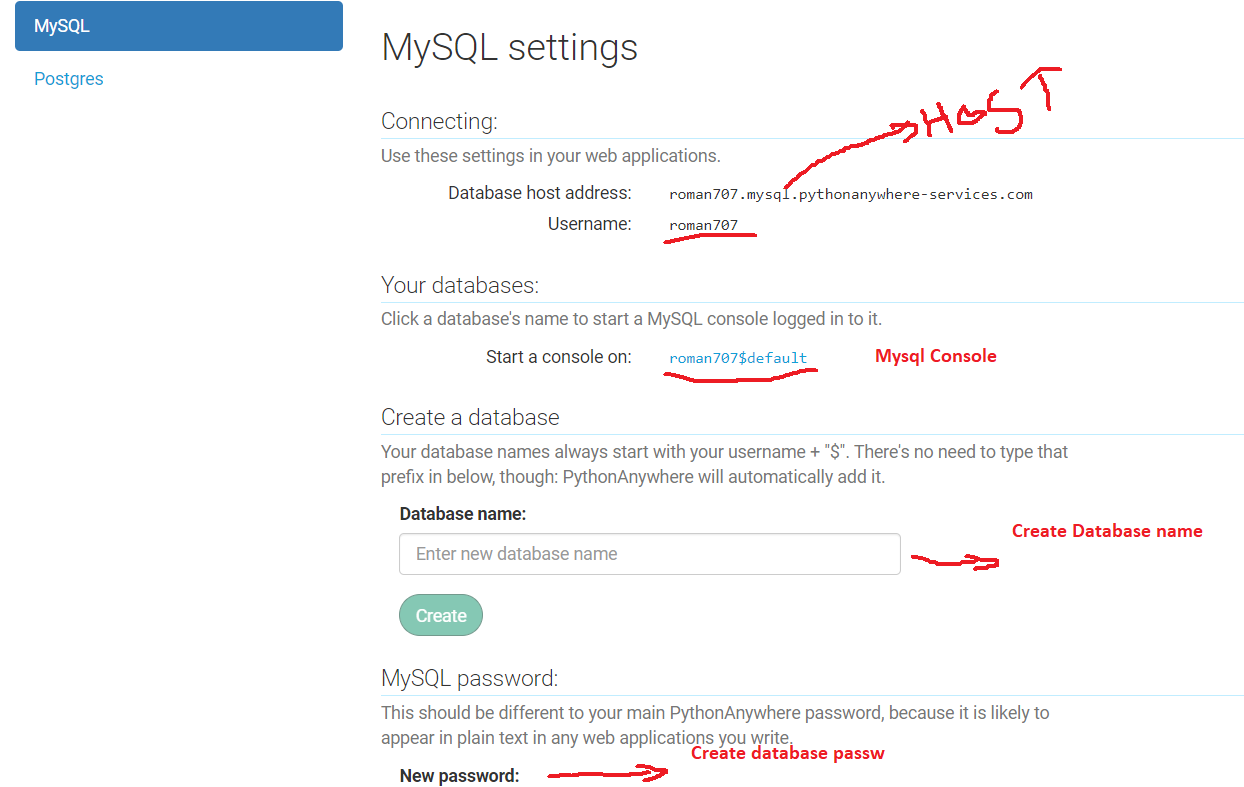
It has created dashboard with user name roman707.

Database:

First we are going to configure MYSQL database for our project so that we can easily set up all file step by step. If your are not using MYSQL or Postgrey database you can skip this configuration steps.

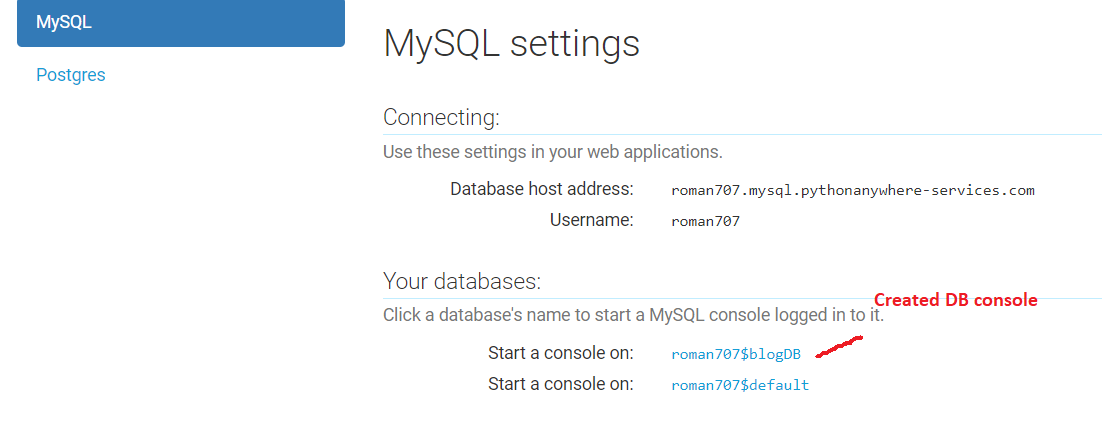
1. Click on database tab
2. Create password to initialize MYSQL database.

You can see here our data base has been initialized and we have been provided Database **Host** and **user\_name** and database **console**.

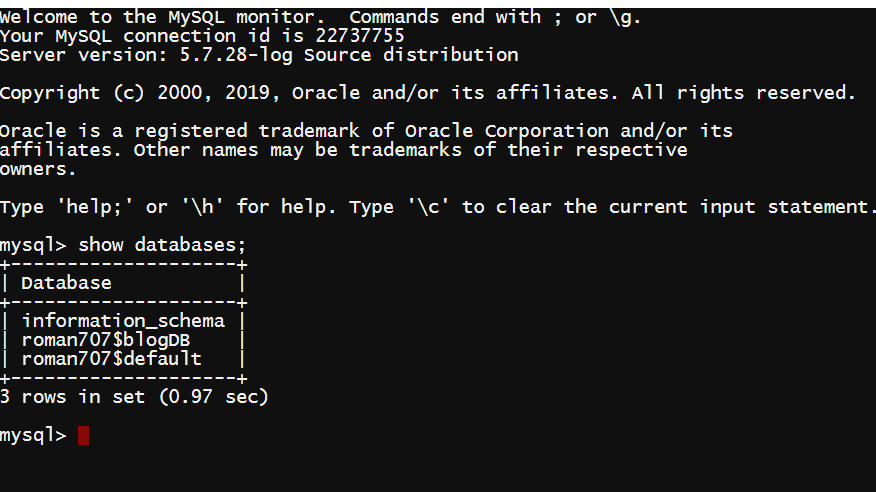


Now we have to create database name and password so that we can update update database configuration in Setting.py. Please note down database name and password which you will create.

Now click MySQL created console (in this case **blogDB**) and type



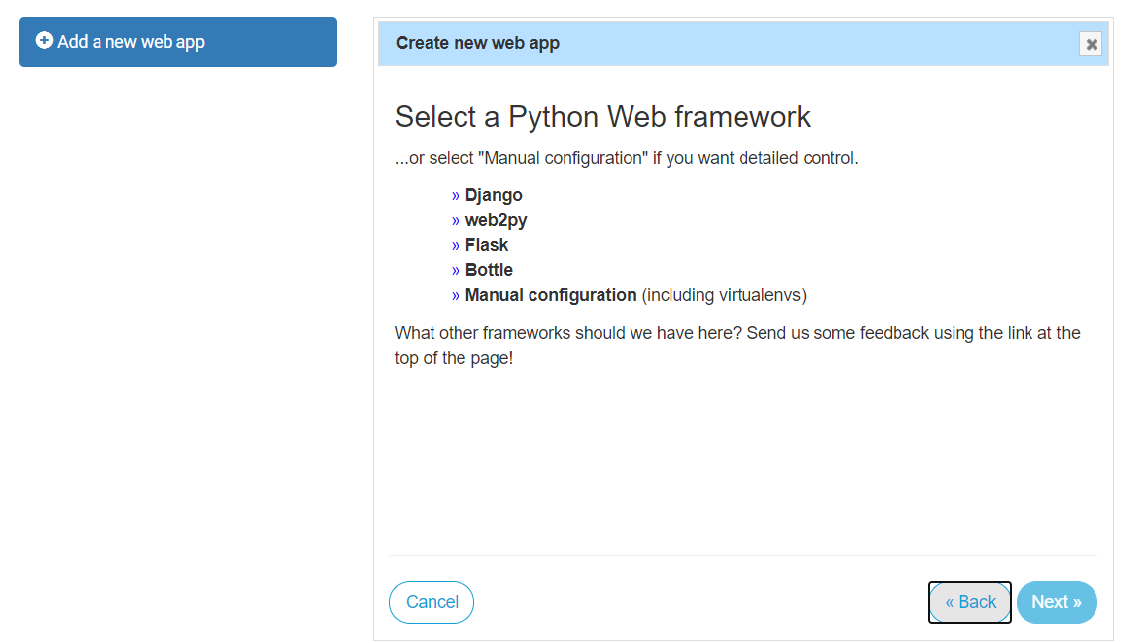
Show databases # to show your databases



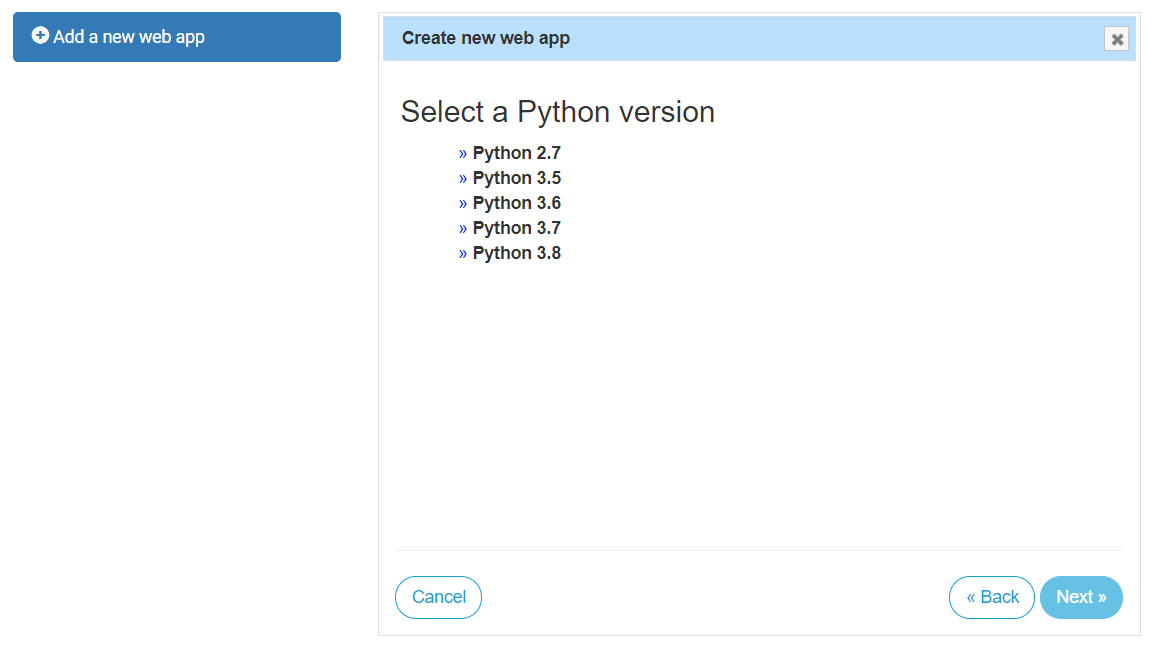
Now We are done with Database Creation and configuration.

**Creating web app**:

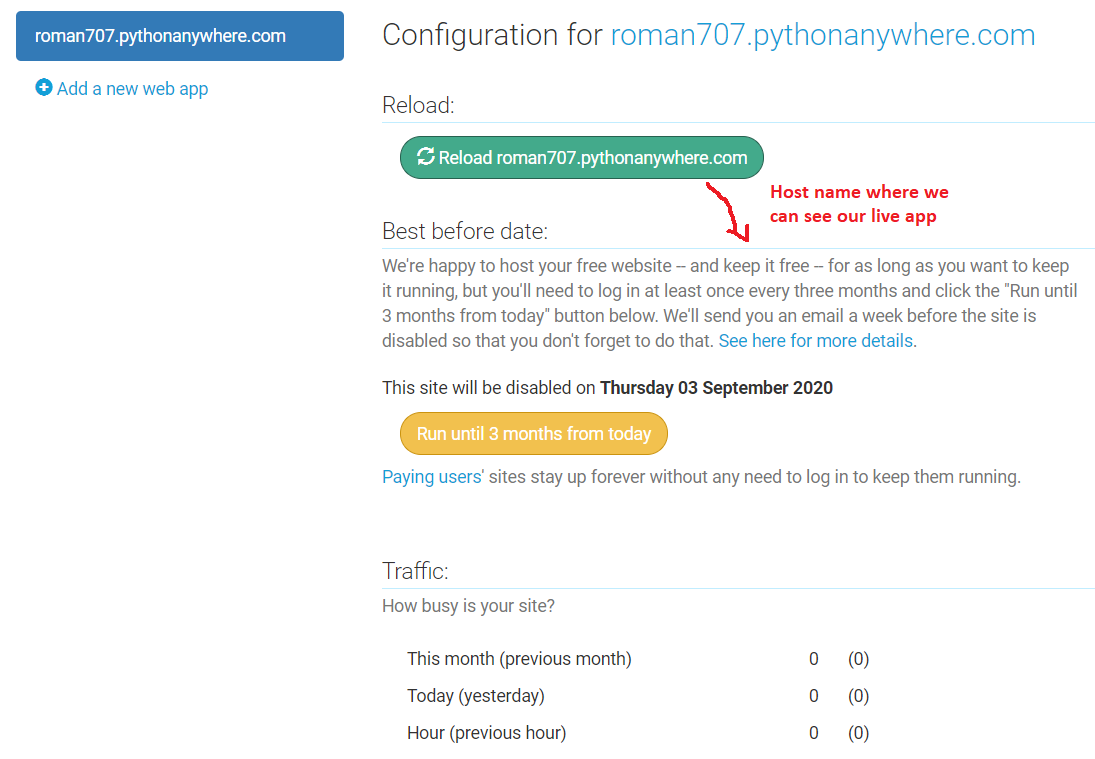
Click on Web tab ,click add new app, click next button then select Manual configuration.



Now Select python version on which your would run you app.



Click next and finally create it.



Remember we have to configure this page in thre places.

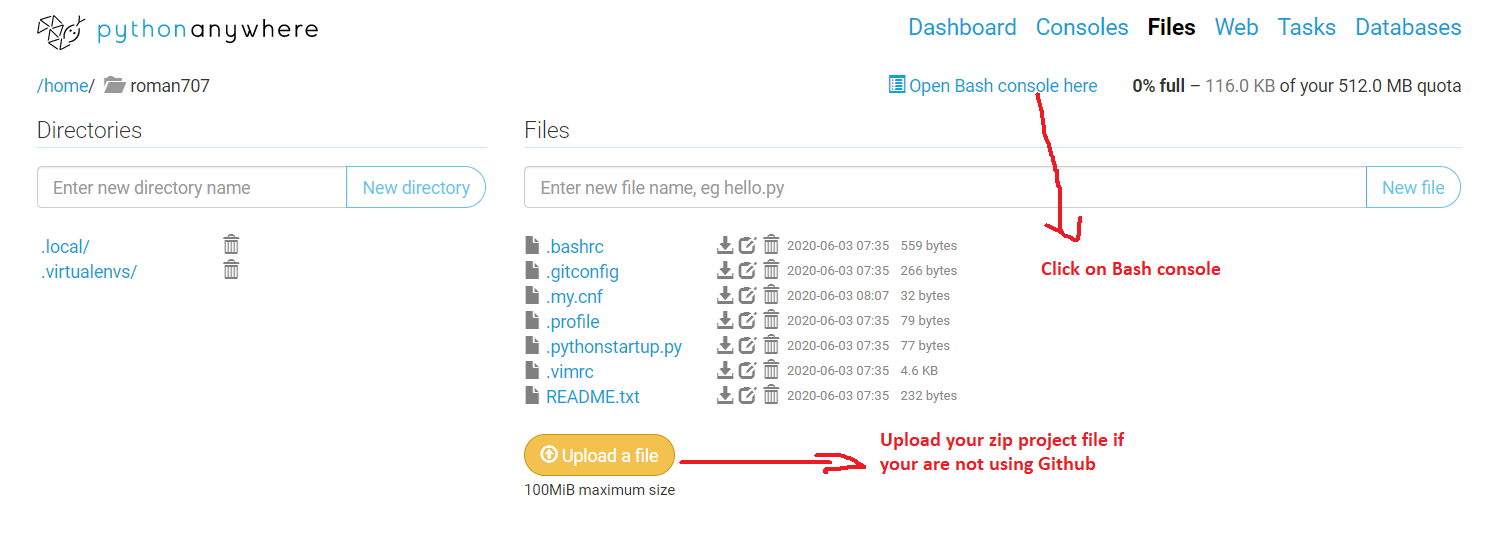
1. Code

WSGI configuration file:

1. **Virtualenv**
2. Static files

Project configuration:

1. Click on File Tab

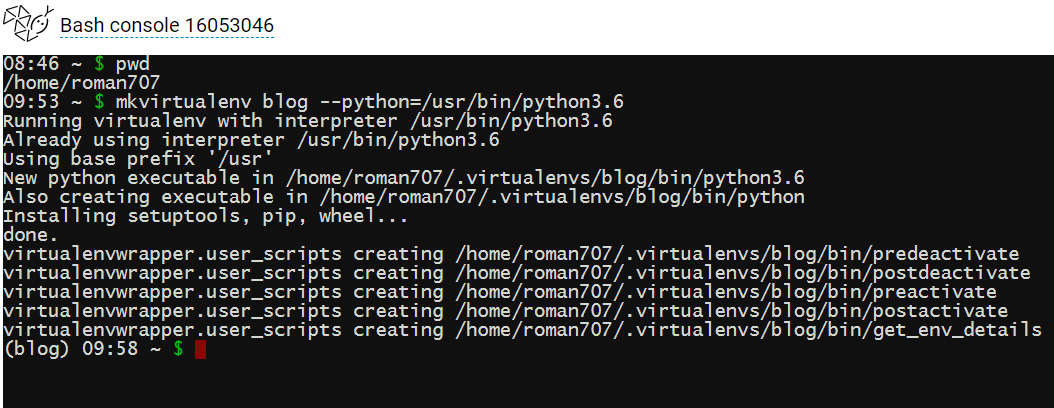


Upload your zip file if you are not using Github and want to upload your zip project. Click on Bash consel, it is a command line interface where we make virtual env and upload our project.

When virtual Env open, type pwd to check current working directory.

Now Create virtual environment, where we put our source code.

mkvirtualenv blog --python=/usr/bin/python3.6



Now we have to run some command in order to setup our project.

1. >> dir # to check list of directory

If you have uploaded a zip file then you need to unzip it

1. >> unzip project\_name.zip

OR you can directly import all latest code from you Github

>> git clone git clone https://github.com/tauovir/myblog.git

1. dir # check list of file and directory

Now get into you project directory (in this case)

1. cd myblog

Since we have created new empty virtual environment, we need to install all libraries or packages which are necessary for running our project.

How we can know which packages are required? Do not worry, you have remembered we have create project\_requirement.txt file in our project. This file contains all required libraries which are essential for our web app.

Type bellow command

Pip install -r project\_requirement.txt

When it install all packages just tyoe last command

Now make some changes in Setting.py file

DEBUG = False

ALLOWED\_HOSTS = ['\*']

# =================Static file ===================

STATIC\_URL = '/static/'

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

# STATICFILES\_DIRS = [

# os.path.join(BASE\_DIR, "static"),

# ]

# Media Files

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR,'media/')

# ==============Database=================

DATABASES = {

'default': {

#'ENGINE': 'django.db.backends.sqlite3',

#'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

'ENGINE': 'django.db.backends.mysql',

'NAME' : 'roman707$blogDB',

'USER' : 'roman707',

'PASSWORD' :'khan1234',

'HOST' : 'roman707.mysql.pythonanywhere-services.com',

'PORT' : '',

'OPTION' : {

'init\_command' : "SET sql\_mode = 'STRICT\_TRANS\_TABLES'"

}

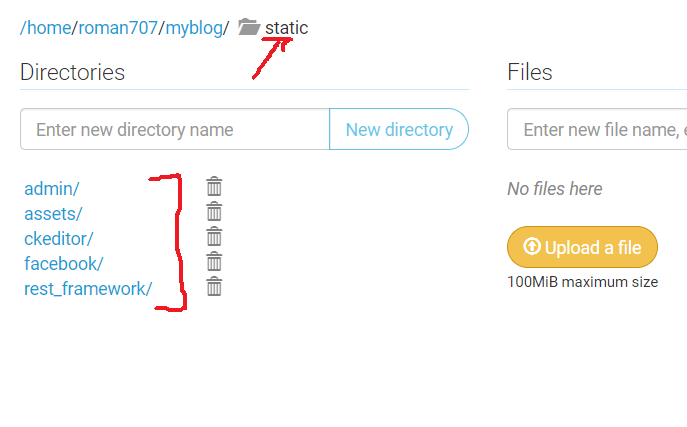
}

}

Now again goto Bash console and type

Python manage.py collectstatic # it will collect all static files

# Check your static folder , you will see all static folder are there

 Now make migrations.

Python manage.py makemigrations

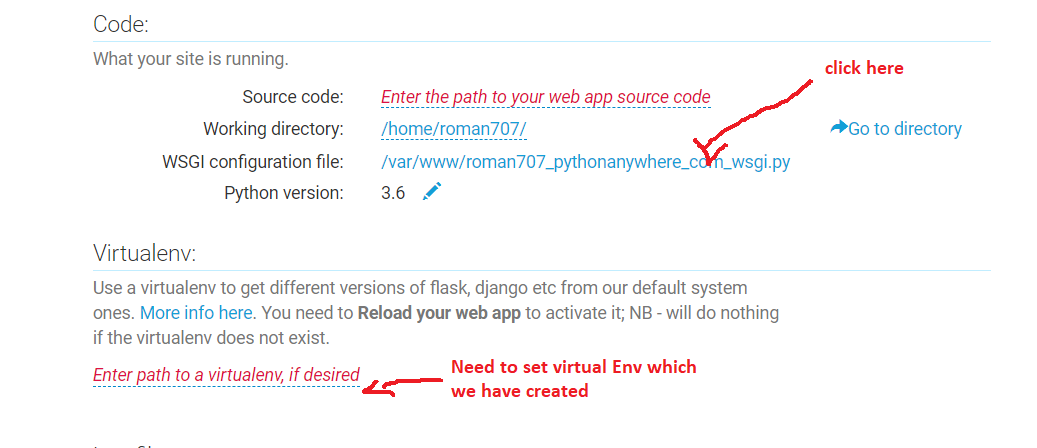
Python manage.py migrate

Now its all done from project site.

Now we need to configure some setting on Web tabs

Click on web tabs and scroll donw to **code section;**

Here we need to configure WSGI configuration file,



This is default configuration file, which run by default on server. Remove all content except Django section’s content; uncomment python code and just replace **mysite** to your actual folder name.

In this case we set “myblog”

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

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

**import os**

**import sys**

**#**

**## assuming your django settings file is at '/home/roman707/mysite/mysite/settings.py'**

**## and your manage.py is is at '/home/roman707/mysite/manage.py'**

**path = '/home/roman707/myblog'**

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

**sys.path.append(path)**

**#**

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

**#**

**## then:**

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

**application = get\_wsgi\_application()**

Save it and come out from there.

Now set virtual environment.

just type your blog name. Since in this case we have created virtual environment ‘blog’. Just simply write in the box. It will automatically detect full path. Else you can write full path of your virtual environment. Example:

/home/username/.virtaulenvs/virtual\_env\_name

[/home/roman707/.virtualenvs/blog](https://www.pythonanywhere.com/user/roman707/webapps/)

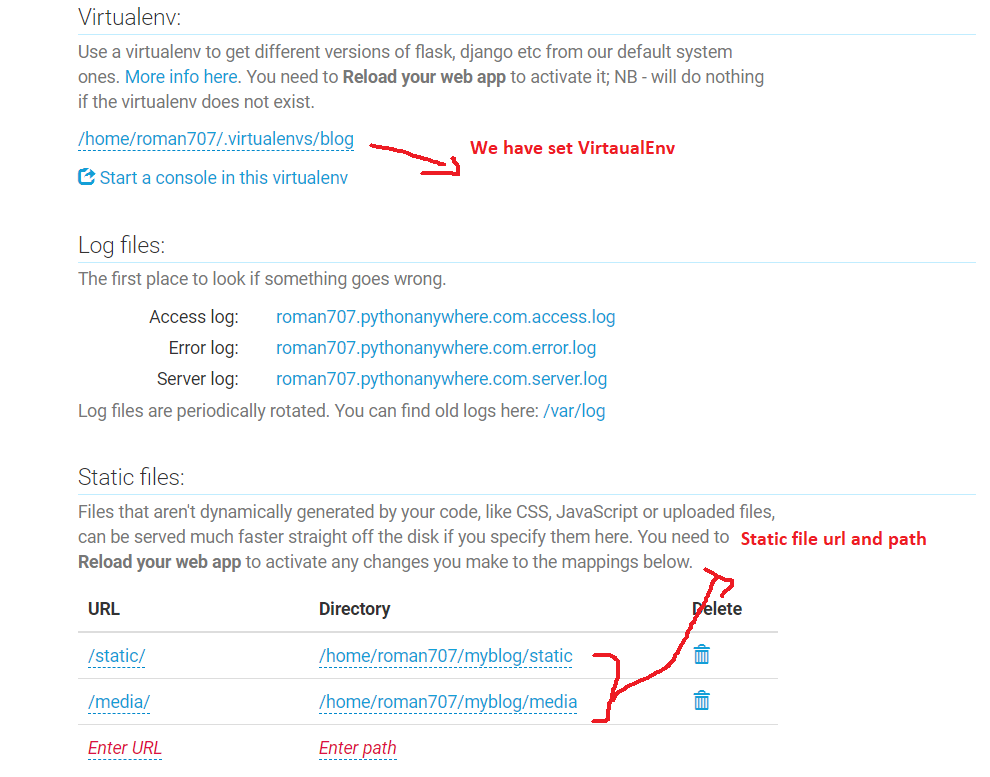
Static file:

Set static url and directory. Server will read all static files from the below directory.

URL Directory

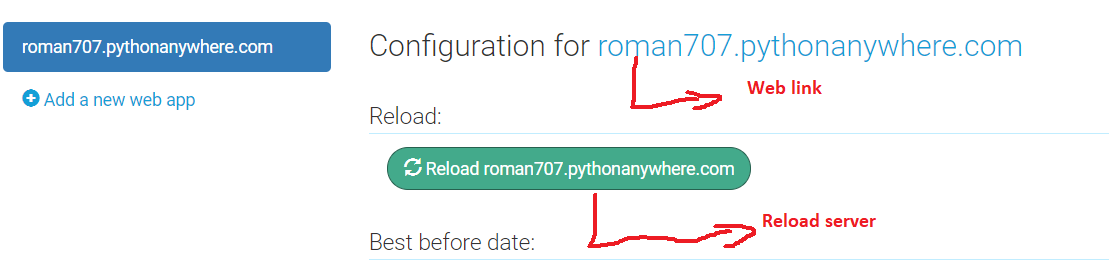
[/static/](https://www.pythonanywhere.com/user/roman707/webapps/) [/home/roman707/myblog/static](https://www.pythonanywhere.com/user/roman707/webapps/)

[/media/](https://www.pythonanywhere.com/user/roman707/webapps/) [/home/roman707/myblog/media](https://www.pythonanywhere.com/user/roman707/webapps/) ( you can copy this path from your file directory.



Now all done

Just reload your server and click on web link;



Live web

