

Hébergement d'un projet Django

Héberger un site Web signifie simplement s'assurer que votre site Web est accessible sur le World Wide Web (WWW). Cela se fait généralement de deux manières. Vous pouvez payer l'hébergement avec un fournisseur de services ou l'héberger sur un site d'hébergement gratuit.

Nous allons utiliser dans ce tutorial le site www.pythonanywhere.com.

PythonAnywhere est un environnement de développement intégré en ligne (IDE) et un service d'hébergement Web basé sur le langage de programmation Python. PythonAnywhere facilite la création et l'exécution de programmes Python dans le cloud.

- 1- Il faut tester votre application Django en local et s'assurer qu'elle ne présente aucun bug.

```
D:\testHebergement\mysite>python manage.py makemigrations
D:\testHebergement\mysite>python manage.py migrate
.....
D:\testHebergement\mysite>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 12, 2020 - 18:23:28
Django version 3.0.5, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

2- Créer le fichier requirements.txt

Afin de pouvoir installer les mêmes bibliothèques que nous utilisons dans notre projet local sur notre environnement de développement, nous allons créer un fichier **requirements.txt** avec pip.

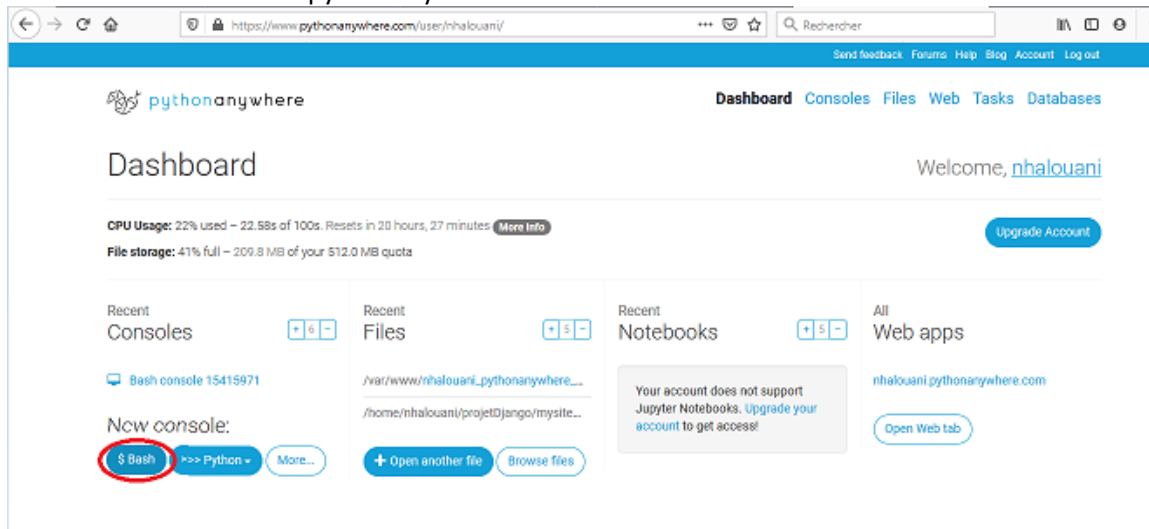
Pour ça, assurez-vous d'activer votre environnement virtuel, et à la racine du projet, utilisez la commande :

```
pip freeze > requirements.txt
```

Votre fichier devrait ressembler à ceci :

```
asgiref==3.3.4
Django==3.2.1
Pillow==8.2.0
pytz==2021.1
sqlparse==0.4.1
```

- 3- Déposer et héberger l'application Django sur Github. (voir le tutorial sur Github). Copier le lien (URL) de l'application sur Github (<https://github.com/nHalouani/projetDjango>).
- 4- Aller sur www.pythonanywhere.com, s'inscrire et se connecter
- 5- Accéder au shell bash de pythonanywhere.



```
17:52 ~ $ git clone https://github.com/nHalouani/projetDjango
14:21 ~/projetDjango $ mkvirtualenv -p python3.8 env
(env) 14:21 ~/projetDjango (master) $ pip install -r requirements.txt
(env) 14:21 ~/projetDjango (master) $ python manage.py makemigrations
..... # pour cordonner le modèle et la base de données
(env) 14:21 ~/projetDjango (master) $ python manage.py migrate
..... # pour créer la base de données
```

- 6- Créer un fichier « .env » dans le même dossier que manage.py et qui contient les lignes suivantes (à copier du fichier setting.py):

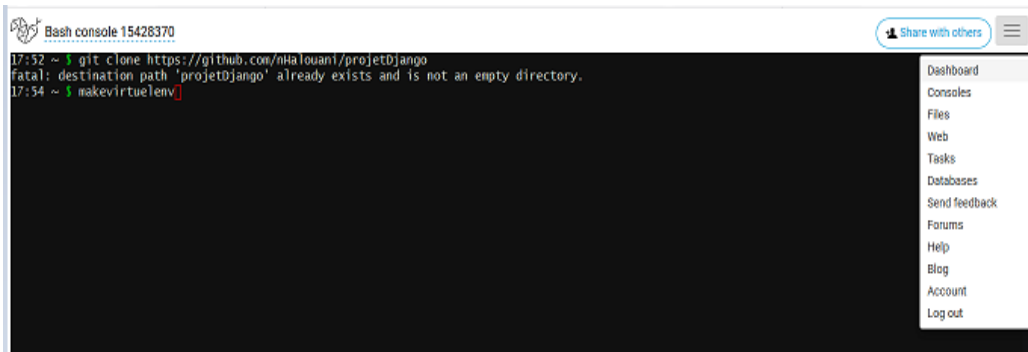
```
SECRET_KEY = '%+52kc=jk_c-q-^9%0@joyqo3-b(&^0=g^d_mm9#=@__t*(btu'
DEBUG = False
ALLOWED_HOSTS = ['zghalOms2023.pythonanywhere.com']
```

- 7- À ce stade, nous allons créer un fichier `.gitignore` qui va contenir les fichiers et dossiers que nous ne souhaitons pas inclure dans notre dépôt Git.

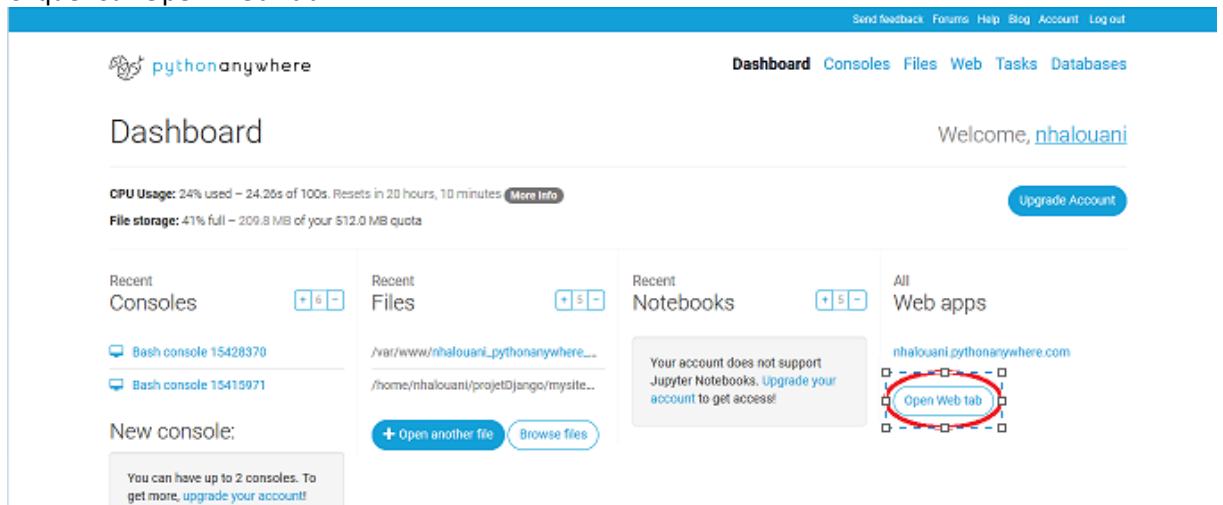
Le fichier `.gitignore` aura le contenu suivant :

```
.idea/  
db.sqlite3  
__pycache__/  
.env/  
.env
```

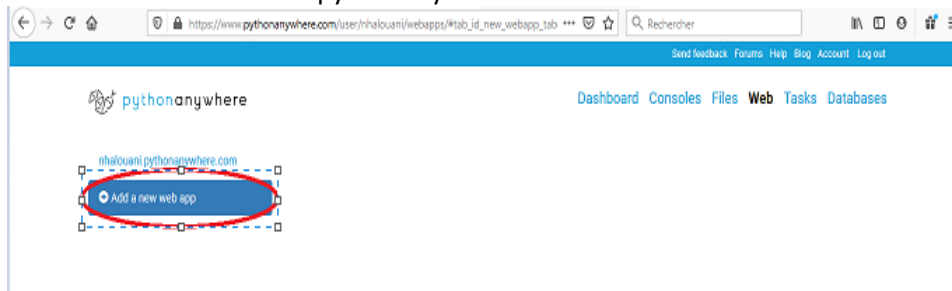
- 8- Basculer vers Dashbord.



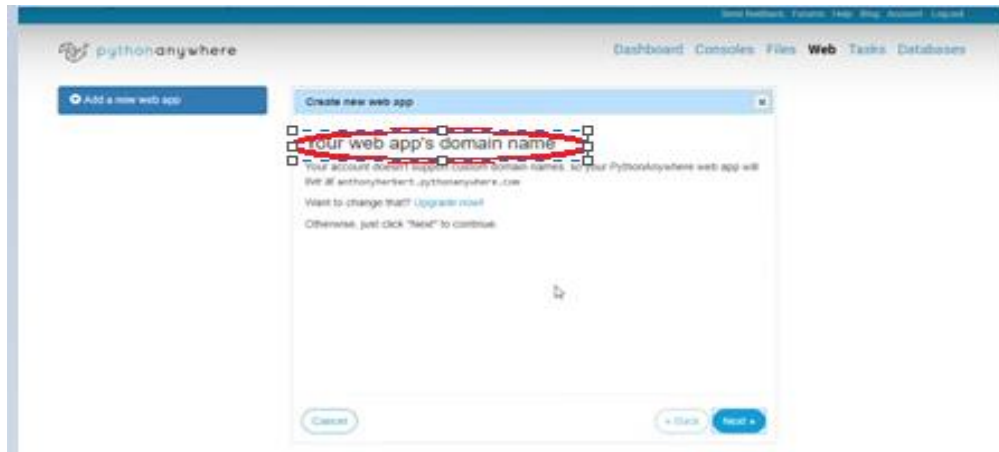
Cliquer sur Open Web Tab



Puis appuyer sur le bouton « **Add a new web app** ». On donne le domaine de l'application qui est dans ce cas `nhalouani.pythonanywhere.com`



Dans l'écran qui suit cliquer sur next



9- Sélectionner « Manuel configuration » dans l'écran suivant



10- Sélectionner la version courante utilisée de python dans l'écran suivant python3.8



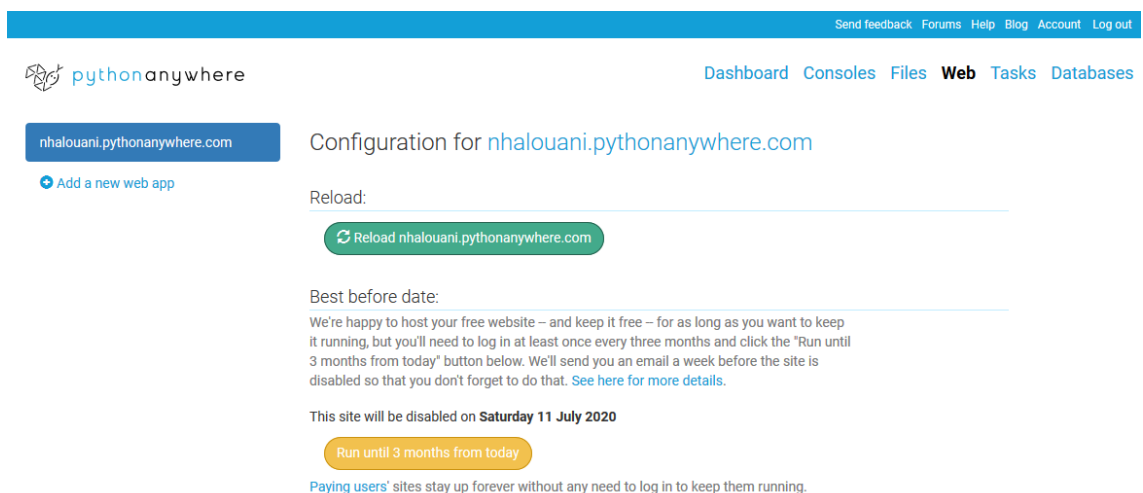
Taper next

11-



Il vous indique dans cet écran qu'il faut reconfigurer le fichier WSGI de l'application et aussi spécifier l'environnement virtuel. Taper next.

Vous voyez apparaitre l'écran suivant :




12- Les paramètres à spécifier pour configurer l'application web sont les suivants :

a- Partie code

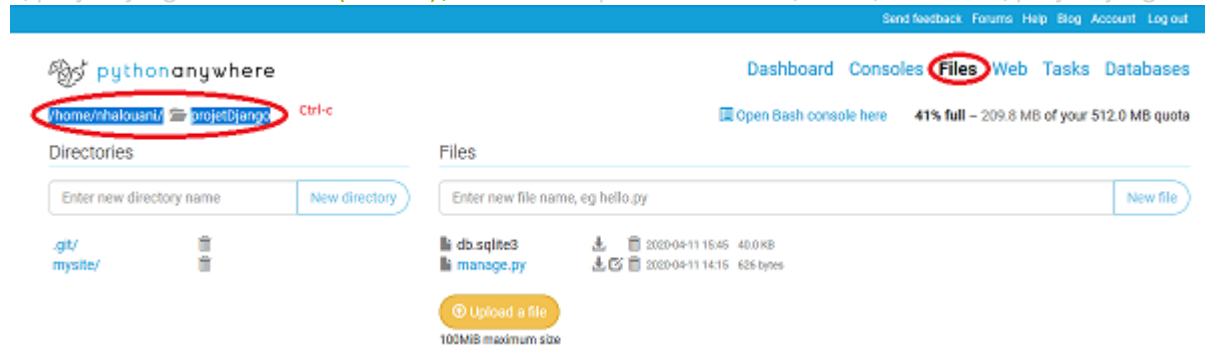
Code:

What your site is running.

Source code: </home/nhalouani/projetDjango>
Working directory: </home/nhalouani/>
WSGI configuration file: /var/www/nhalouani_pythonanywhere_com_wsgi.py
Python version: 3.7 

Les chemins du code source et du répertoire de travail, on peut les obtenir soit à partir de la console du shell (commande pwd), soit de l'explorateur de fichier de **pythonanywhere** ainsi :

```
(envbis) 11:18 ~/projetDjango (master)$ pwd /home/nhalouani/projetDjango (envbis) 11:18  
~/projetDjango (master)$ pwd /home/nhalouani/projetDjango
```



Ou bien sur la console écrire :

```
(env) 11:18 ~/projetDjango (master)$ pwd  
/home/nhalouani/projetDjango
```

b- Maintenant nous allons modifier le contenu du fichier :
`/var/www/nhalouani_pythonanywhere_com_wsgi.py`

Dans le fichier « nhalouani_pythonanywhere_com_wsgi.py » il faut dé-commenter le code relatif à django comme suit :

```
# ++++++ DJANGO ++++++  
# To use your own django app use code like this:  
import os  
import sys  
## assuming your django settings file is at '/home/nhalouani/mysite/mysite/settings.py'  
## and your manage.py is at '/home/nhalouani/mysite/manage.py'  
path = '/home/nhalouani/projetDjango'  
if path not in sys.path:  
    sys.path.append(path)  
os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite.settings'  
## then:  
from django.core.wsgi import get_wsgi_application  
application = get_wsgi_application()
```

Virtualenv:



Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/nhalouani/.virtualenvs/env](#)

d- Indiquer le chemin du répertoire statique qui contient les fichiers .HTML.

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, 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.

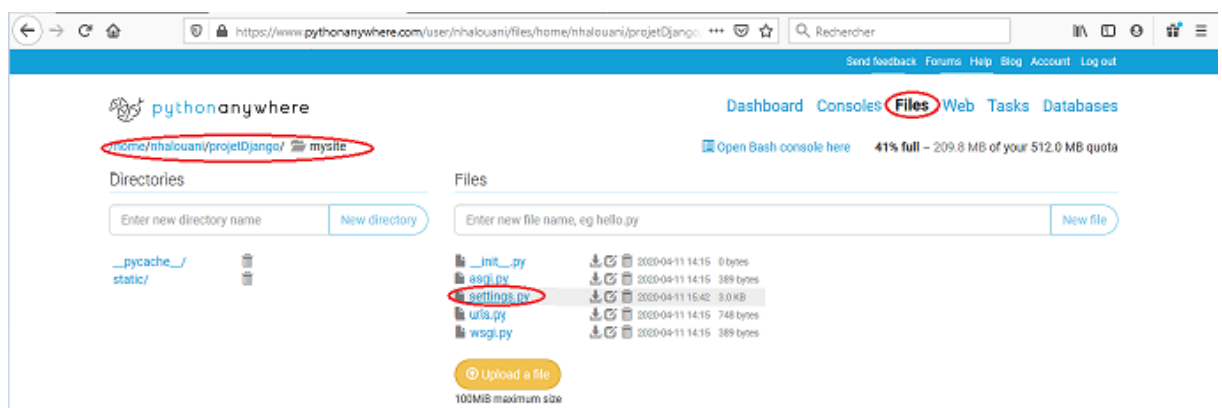
URL	Directory	Delete
/static/	/home/nhalouani/ProjetDjango/static/	
/media/	/home/nhalouani/ProjetDjango/media/	

Il ne nous reste plus qu'à collecter tous les fichiers statiques de notre application dans le dossier correspondant à STATIC_ROOT avec la commande `python manage.py collectstatic`.

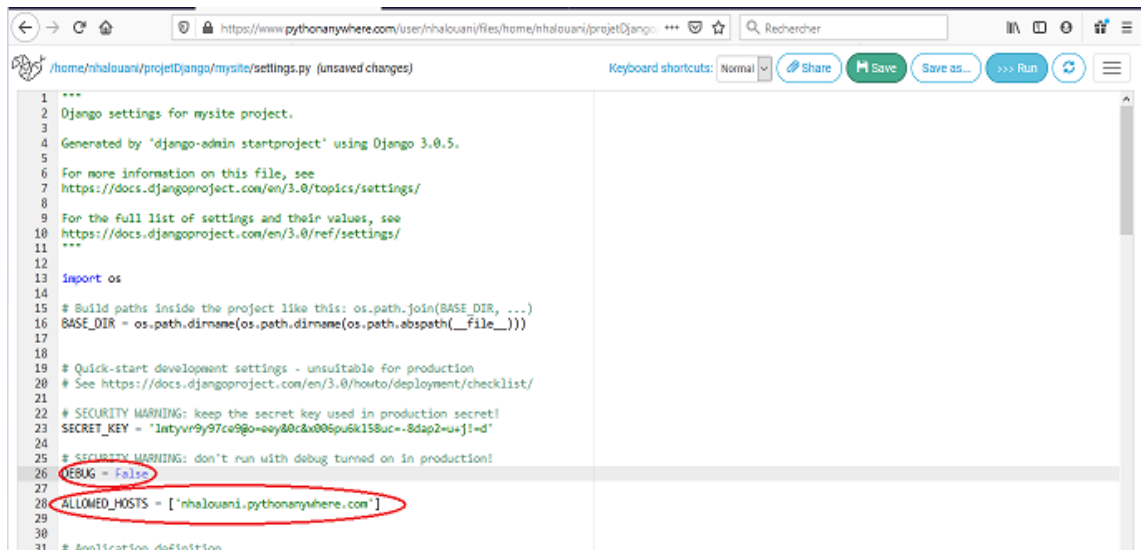
De retour dans notre console, dans le dossier projetDjango :

```
(env) 11:18 ~/projetDjango (master)$ python manage.py collectstatic
```

e- Modifier le contenu du fichier « setting.py » de notre application pour y intégrer le nom de domaine.

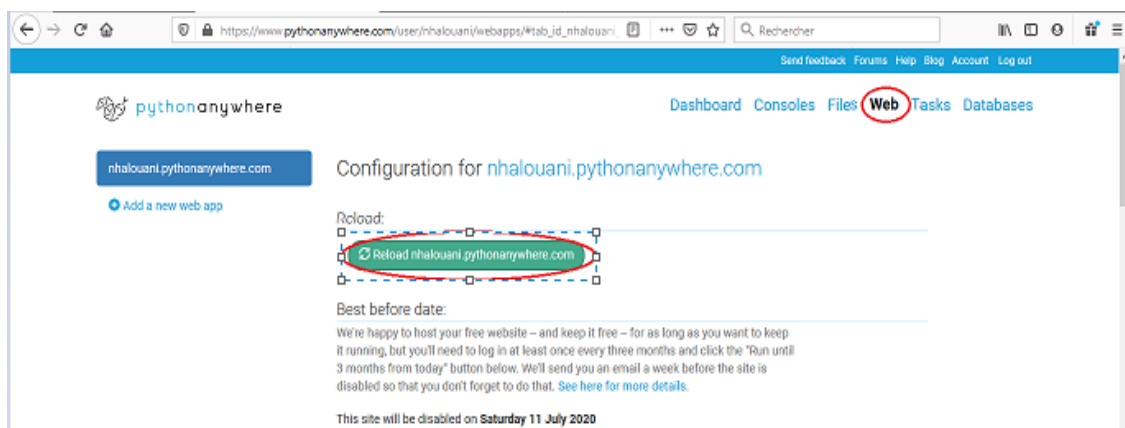


Puis modifier le fichier en ajoutant le domaine dans ALLOWD_HOSTS=[] et mettre DEBUG à False.

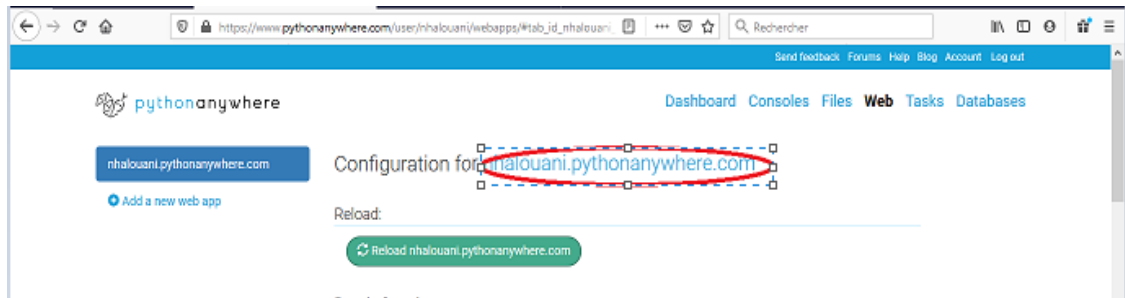


```
1 ***
2 Django settings for mysite project.
3
4 Generated by 'django-admin startproject' using Django 3.0.5.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.0/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.0/ref/settings/
11 ***
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18 # Quick-start development settings - unsuitable for production
19 # See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
20
21 # SECURITY WARNING: keep the secret key used in production secret!
22 SECRET_KEY = 'Intyvr9y97ce9g0=00y80c&x005pu6k158uc=-8dap2=+j!d'
23
24 # SECURITY WARNING: don't run with debug turned on in production!
25
26 DEBUG = False
27
28 ALLOWED_HOSTS = ['nhalouani.pythonanywhere.com']
29
30
31 # Application definition
```

13- Une fois toutes les configurations effectuées, nous devons recharger l'application.



Puis pour lancer votre application cliquer sur le nom de domaine comme suit :



Si tout va bien vous allez voir apparaître l'écran d'accueil de l'application, s'il y a une erreur vous pouvez consulter son origine dans le fichier « ...error.log »

Log files:

The first place to look if something goes wrong.

Access log: [nhalouani.pythonanywhere.com.access.log](#)

Error log: [nhalouani.pythonanywhere.com.error.log](#)

Server log: [nhalouani.pythonanywhere.com.server.log](#)

Log files are periodically rotated. You can find old logs here: [/var/log](#)

L'accès à ce fichier se fait ainsi :

