# Blog- Django – For RE

# Checklists

Installations and environment on windows:

1. Python – done
2. Code editor – sublime text3 – done
3. Virtual env setup and installing Django 2.2.4 - Done

* mkdir djangogirls
* C:\Users\Name\djangogirls> python -m venv myvenv

Activate - myvenv\Scripts\activate

* Upgrade pip => python -m pip install --upgrade pip
* Create requirements.txt with [Django~=2.2.4]
* pip install -r requirements.txt

1. Installing Git - done
2. Making an account on Github - done
3. Create a python anywhere account – not done

Next for Project

1. Create the structure
   1. django-admin.exe startproject mysite .

manage.py is a script that helps with management of the site. With it we will be able (amongst other things) to start a web server on our computer without installing anything else.

The settings.py file contains the configuration of your website.

Remember when we talked about a mail carrier checking where to deliver a letter? urls.py file contains a list of patterns used by urlresolver.

Let's ignore the other files for now as we won't change them. The only thing to remember is not to delete them by accident!

B change settings [ lang, time zone etc]

* 1. python manage.py migrate
  2. python manage.py runserver // To run the server

1. Now configure project blog
2. python manage.py startapp blog
3. We need to find INSTALLED\_APPS and add a line in the settings.py file containing 'blog.apps.BlogConfig', just above ]

Now in blog/models.py

from django.conf import settings

from django.db import models

from django.utils import timezone

class Post(models.Model):

author = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE)

title = models.CharField(max\_length=200)

text = models.TextField()

created\_date = models.DateTimeField(default=timezone.now)

published\_date = models.DateTimeField(blank=True, null=True)

def publish(self):

self.published\_date = timezone.now()

self.save()

def \_\_str\_\_(self):

return self.title

Then run this command

python manage.py makemigrations blog

Django prepared a migration file for us that we now have to apply to our database. Type python manage.py migrate blog and the output should be as follows:

How I solved the github issue

Created ssh key and pushed the code as below

1. Open git bash in the same folder
2. ssh-keygen
3. keep doing enter
4. a folder would be created C:\Users\Nine Latitude\.ssh
5. open id\_rsa.pub file in notepad or sublime
6. copy the key and upload it to the deploy key section of settings of the project
7. back to bash key
8. run
9. git remote add origin <https://github.com/rubykumari1/RealEstate_Blog.git>
10. git branch -M main
11. git push -u origin main

Now on pythonanywhere open bash terminal going to dashboard

Step1:

pip3.8 install --user pythonanywhere

Step 2

pa\_autoconfigure\_django.py --python=3.8 <https://github.com/rubykumari1/RealEstate_Blog.git>

Step3

python manage.py createsuperuser

Now after adding the html templates deploy on github

git status

git add .

git status

git commit -m "Changed the HTML for the site."

git push

On python anywhere to run the following commands

To check if you are in your project directory

cd ~/<your-pythonanywhere-domain>.pythonanywhere.com

git pull