Django

1. Set up Environment
   1. **python –V** : check python version
   2. **pip install pipenv** : install pipenv
   3. **pipenv –version** : check pipenv version
   4. **pipenv shell** : launch virtual environment
   5. **pipenv install Django** : install Django
   6. **pip show Django** : shows details of Django
2. Start Project
   1. **django-admin startproject mysite .** : Creates site packages in ‘mysite’ folder
   2. **python manage.py runserver 0.0.0.0:8000** : creates server
   3. Upon getting warning about unapplied migrations, you should:

**python manage.py migrate**

**python manage.py runserver 0.0.0.0:8000**

* 1. In browser, open url: “localhost:8000”
  2. To go to admin page, use “localhost:8000/admin”
  3. Setup username and password for Django:
     1. Go to Terminal
     2. Type: **python** **manage.py createsuperuser**
     3. Enter Username, Email and Password
     4. Run server: **python manage.py runserver 0.0.0.0:8000**
  4. Create App:
     1. **Python manage.py startapp feed** : creates app instance. “feed” is the name of the folder that gets created.
     2. Open file “settings.py” in mysite folder.

In Installed Apps section, add name of the app created (in this case it is ‘feed’)

* 1. Create Database Model:
     1. Open the models.py file from ‘feed’ folder.
     2. Create class

Example:

**Class Post(models.Model):**

**Text = models.CharField(max\_length = 140, blank = False, null = False)**

Save File.

* + 1. Commit migrations by passing command in terminal:

**python manage.py makemigrations**

See the file generated

Then run code:

**python manage.py migrate**

* + 1. Go to “feed/admin.py” to register your model

Import the model (example in this case, import statement will be:

**from .models import Post** )

create PostAdmin class as below:

**class PostAdmin(admin.ModelAdmin):**

**pass**

link Post and PostAdmin together. This will register Post database model.

**admin.site.register(Post, PostAdmin)**

* + 1. Re-run the Django server and check the Post database visible there.

If you create a new post, it will show its name as **“Post object”**. This is because Django treats every line of database as an Object.

To change the name of this entry, we do the following:

Open file: “feed/models.py”

In class Post, enter:

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

return self.text

This will show the entry with the actual text entered in that entry.

1. Setting View:
   1. Create a file called “urls.py” in “feed” folder.
   2. Create a list of patterns/paths. URL patterns will be a list containing the path. Path is given in format **path(<1. url address>, <2. View Rendered>, <3. Name of the page>)**.

Example: path(“”, renderview, name=”index”)

* 1. The file should be similar to the file found in “mysite/urls.py”
  2. You can write class based and function based views. But professional way is to use class based views.
  3. Create a view (in this case, we will create HomePageView) in file views.py

**from django.views.generic import TemplateView**

**class HomePageView(TemplateView):**

**template\_name= “home.html”**

Django will search for “home.html” file.

* 1. Go back to **urls.py** file in the same folder (“feed/urls.py”) & import the view that is created in the **views.py**

**from .views.py import HomePageView**

**urlpatterns = [path,(“”, HomePageView.as\_view(), name=”index”)]**

Go to file “mysite/urls.py” and add code:

**from django.conf.urls import include**

**from feed import urls as feed\_urls**

**urlpatterns = [ path(‘admin/’, admin.site.urls),**

**path(“”, include(feed\_urls, namespace=”feed”))]**

1. Setting up Templates:
   1. Go to “mysite/settings.py”, scroll down to **templates>DIRS**

Add code:

**import os**

…

**TEMPLATE\_DIR = os.path.join(BASE\_DIR,”templates”)**

**…**

**TEMPLATES = […’DIRS’:[TEMPLATE\_DIR],…]**

* 1. Create folder called **templates**.

This will have html pages like “home.html”

Create file “templates/home.html” & its contents.

You can use the {% extends filename.html %} property to inherit the html code from another file (example: **{% extends header.html %}**