***Activate***

Flask already install:

cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\my\_environments\vitu

Django already install:

cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\my\_environments\django

Scripts\activate

cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro

cd D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_full\_stack

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro>django-admin startproject first\_project

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro>cd first\_project

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\first\_project>mkdir apps

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\first\_project>cd apps

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\first\_project\apps>python ../manage.py startapp first\_app

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\first\_project\apps>cd first\_app

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\first\_project\apps\first\_app>nul>urls.py

Access is denied.

(django) D:\A\_learning\abc\codingdojo\_week3\_python\python\_stack\django\django\_intro\first\_project\apps\first\_app>

your\_project\_name\_here/your\_project\_name\_here/settings.py

INSTALLED\_APPS = [

'apps.first\_app', <= # added this line. Don't forget the comma!! remember to change name

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

your\_project\_name\_here/your\_project\_name\_here/urls.py

from django.conf.urls import url, include # added an import!

# from django.contrib import admin # comment out, or just delete

urlpatterns = [

url(r'^', include('apps.first\_app.urls')), # use your app\_name here <= remember to change name

# url(r'^admin/', admin.sites.urls) # comment out, or just delete

]

My second way:

from django.urls import path, include

urlpatterns = [

    path('', include('apps.ninja\_farm.urls')),

]

your\_project\_name\_here/apps/your\_app\_name\_here/urls.py

from django.urls import path

from . import views

urlpatterns = [

    path('', views.index),

    # path('bla', views.farming),

    # path('blabla', views.reset),

]

your\_project\_name\_here/apps/your\_app\_name\_here/views.py

from django.shortcuts import render, HttpResponse

def index(request):

    return HttpResponse("i love youuuuuuuuuu")

remember to change link in form in html and etc…

add this if there static in html:     {% load static %}

change app\_name for link and script in static

add this for hidden:

              {% csrf\_token%}

Now, in a new Django project, open up your models.py file and put the following model there:

class Hogwarts(models.Model):

    name = models.CharField(max\_length=45)

    house = models.CharField(max\_length=45)

    pet = models.CharField(max\_length=45)

    year = models.IntegerField()

Once you have the model, run the two migration commands in your terminal and open the shell:

python manage.py makemigrations

python manage.py migrate

python manage.py shell

Once we're in the shell, we can access all of our functions and classes in our files. To do so, we just need to specify which modules (files) we need. Since we are interested specifically in working with our models, let's import them:

>>> from apps.your\_app\_name\_here.models import \* (or model u want to import)

**Caution:**  Since models.py contains the classes you wrote, importing everything in models.py is okay, but generally when you're importing other libraries/modules, the Django community discourages the practice of importing all (\*).

**Helpful Tip**

You've probably noticed in the shell that printing whole objects just results in something like <Movie Object (1)>, which is not particularly helpful. To change how our models display, we can *override* the \_\_repr\_\_ method in the class. This is pretty handy and shows how we can leverage some of Python's magic methods to make our lives easier.

class Movie(models.Model):

# fields removed for brevity

*def* \_\_repr\_\_(self):

return f"<Movie object: {self.title} ({self.id})>"

**chmod 400 rannicy.pem**

**ssh -i "rannicy.pem" ubuntu@ec2-13-58-12-144.us-east-2.compute.amazonaws.com**

sudo ln -s /etc/nginx/sites-available/belt /etc/nginx/sites-enabled

$ git remote -v

# View current remotes

> origin https://github.com/*OWNER/REPOSITORY*.git (fetch)

> origin https://github.com/*OWNER/REPOSITORY*.git (push)

> destination https://github.com/*FORKER/REPOSITORY*.git (fetch)

> destination https://github.com/*FORKER/REPOSITORY*.git (push)

$ git remote rm destination

# Remove remote

$ git remote -v

# Verify it's gone

> origin https://github.com/*OWNER/REPOSITORY*.git (fetch)

> origin https://github.com/*OWNER/REPOSITORY*.git (push)

If you are sure that the directory exists, then:

(sudo) rm -rfv /path/

To delete the entire directory to your folders and files