Teminal:

Cd desktop

Django-admin startproject (\_name of project\_)

Cd (name of fproject)

Python3 manage.py runserver

Brings up a local address to load a website.

How to create an app inside the project:

Go into terminal

Python3 manage.py (\_\_name of orginal app\_\_) (\_\_name of app you want to create)

Have to register app in orginal app setting .py folder

Go to folder, in folder go to “installed apps”

Add the exact name of the app(new created app)

Everytime we initialize new app we do this

Creating App Level URL Router.

Create new file in new app made named urls.py

Copy ‘From Django.urls import path’

Copy all of ‘urlpatterns’ and set everything inside [ ] to blank

In orginal app urls.py at the ‘from dJango.urls import path,’ add “include”

Then inside urlpatterns add ‘path(‘ ‘,include ((‘name of app created’).urls),

In urls.py in app created, we add in urlpatterns, ‘path(‘ ‘, views.home, name = ‘home’

In same folder, import under the imported path, ‘From .import views)

In views.py in app created we add (under import render) ‘from django.http import httpRespomse.

Same folder:

def home(request):

return HTTPResponse(‘Hello World’)

We should be able to see “Hello World” on our website.

How to Load Templates

In urls.py folder we will enable a new path for the user by:

‘path(‘about/’, views.about, name=about)

Then define the ‘about’ function

New Line:

def about(request):

return render(request, ‘(name of 2nd app created)/about.html’)

Click on app created orginal folder and create a new folder name ‘templates’ then create another folder inside templates named ‘about.html’

Inside the ‘about.html’ folder, create a H1 tag and put ‘About page’ inside.

This should create an about page

ADMIN PANAL

In terminal:

Python 3 manage.py createsuperuser

Follow instructions for username and password.

Now the login information for the admin page should let you in to the admin panal.

In app created models.py file:

Class Car(models.model):

Name = models.Charfield(max\_length=200)

Go to admin.py :

Uder import admin ‘from’:

from .models import Car

admin.site.register(Car)

We will now see a table named Cars in our admin folder.

To make it clickable we have to apply the changes to our database by going to the terminal

python3 manage.py makemigrations

Then python3 manage.py migrate

python3 manage.py runserver 🡪new server with all saved changes should pop up(url).

To see the actual data we go into models.py (in app created):

Under class:

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

return self.name

The list item should now pop up with the specific name.

HOW TO SHOW ENTREES ON OUR IP WEBSITE.

In views.py:

from .models import Car

Under ‘def about(request):

cars = Car.object.all()

In contex{ }:

‘cars’ : cars

Go into Template file under about.html folder we create a unordered list <ul>

A list <li> inside of it and write a for loop using a ginger template.

<ul>

{% for car in cars %}

<li>

Name of Car: {{ car. name}}. <-- because in models.py we named the models.CharField varaiable ‘name’ we use this 🡪

<li>

{% endfor %}

<ul>

This lets us not go into admin, add (in this example) a car name then it will show up on our local page (ip address) as “Name of Car: (car name we added in admin)”

URL spawns to 🡪 Views.py

Admin is directly linked to

WSGI.py for twitter clone:

Models.py: where the data is created. We create it in the table format. Coloum name row name provided in models

Views.py: what to do with that data

First image> it is a file. So I have ti svae it to separate cloud