**Hello world website with Django**

Create the root directory.

Create the project root directory as a folder on the desktop, name is django\_practice, right click and open with vscode or open the root directory with the following command inside terminal: mkdir django\_practice

Create the virtual environment.

Cd into django\_practice directory and create the virtual environment.

Create a virtual environment using the venv module with the command python -m venv .venv or pip install virtualenv

The name of the virtual environment directory is .venv and the period that precedes the virtual environment directory indicates that the directory will be hidden.

Activate the virtual environment

Inside the django\_practice directory use the following command to activate the virtual environment: .venv/scripts/activate

Or using the gitbash terminal use the following command: source .venv/scripts/activate

Install Django

With the virtual environment activated and at the root of django\_practice directory, use the following command to install Django package: pip install django

Start a new project

With the virtual environment activated and at the root of django\_practice directory, use the following command to start a new project and name the new project django\_project: django-admin startproject django\_project .

The period after the django\_project name is intentional; this will make Django to place all the files inside one single directory.

Apply migrations

When Django created the django\_project directory, several built-in apps were created along .

We need to apply migrations so that these changes can be applied to the database.

With the virtual environment activated, and at the root of django\_practice directory, use the following command to apply migration: python manage.py migrate

Start the development server

With the virtual environment activated and at the root of django\_practice directory, use the following command to start the development server: python manage.py runserver

Stop the development server with the following command: ctrl + c

Starting an app.

With the virtual environment activated and at the root of django\_practice, use the following command to start an app named pages: python manage.py startapp pages.

Add the pages app to the list of installed apps inside the django\_project directory settings

Create a view

Open the view.py file of pages app and write the following codes

from django.http import HttpResponse

def new\_home\_page(request):

return HttpResponse(“Hello World”)

Create a urls.py file inside the pages app

we need to match a url path to this view function.

create a urls.py file inside the pages app and write the following code

from django.urls import path

from .views import new\_home\_page

urlpatterns = [

path( “”, new\_home\_page),

]

update the urls.py (gateway to other urls) file inside the root directory of django\_project

There is an built-in urls.py file inside the root directory django\_project

Open the urls.py file and import the following code:

From django.urls import path, include

Add the pages app url path to the list of the url inside the urls.py file of django\_project directory

The urls.py file has an inbuilt admin url path

urlpatterns = [

path(“admin/”, admin.site.urls),

Path(“”, include(“pages.urls”)

]

Start the development server with python manage.py runserver.

This will display hello world

.GITIGNORE FILE

At the root of django\_practice, create a file and name it .gitignore.

This will be used to store the virtual environment, to prevent it from been pushed to github.

Open the .gitignore file and add .venv

Storing the installed packages of the virtual environment

Record of installed packages inside virtual environment

With virtual environment activated at the root of django\_practice,use the following command to store the packages inside a requiremens.txt file.

pip freeze> requirement.txt

pushing to github account

stop the development server with ctrl + c and at the root of django\_practice directory, deactivate the virtual environment with the following command: deactivate

go to github account and create a repository named django\_pracitce

At the root of django\_practice directory, use the following command to initialize git and push the code to github

git init

git add -A

git commit -m “first commit”

git branch -M main

git remote add origin https://github.com/trevor4ivan/django\_practice.git

git push -u origin main