**First app in Django part 1 (create it in VSCode)**

In our previous lecture we have seen how to setup a lightweight server using Django and now let’s create our first web application. ­There are 2 types of web applications 1)static pages 2)dynamic pages.

* Static pages: everyone will see the same page ex: Wikipedia. We can use HTML, CSS, JavaScript for the creation of static pages.
* Dynamic pages: everyone has their own interface or page or feed ex: facebook, banking sites etc.. these pages need to do some processing.

Let’s create a simple/basic web application of course, dynamic one – print ‘Hello world’ on the page that we have from the previous lecture. We need to use an ***IDE*** to do this, we can use Notepad as well as an IDE to type the code, we can use sublime. Before that we will observe the files that are available in our project.

We are going to use VSCode 🡪 open the projects folder which we have created 🡪 open telusko project 🡪 we should see the files as shown below.

Graphical user interface, text, application

Description automatically generated

Manage.py : useful to manage your project

Settings.py : deals with the settings of the project, we can see that it has a secret key used for deployment, debug will give you lot of information when you get stuck somewhere which is important. Ex: if you are building a project and you want to know what is happening behind the scene, just make DEBUG = True and it will give all the information. But make sure when you are deploying the code on the server set it to False, someone can hack the machine and can see all the logs. Installed apps will show the list of all the apps that are already installed. Middleware for security purposes. Discuss about templates. Databases: when you are creating a web application you also want a database that is supporting it. It can be MySQL, Postgresql

Urls.py : we can see that there is only one url which is for admin

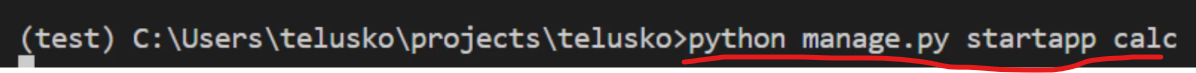
Let’s create our first web application. We have a project right, why we have to create an app. In Django – what we do is we have a project, let’s say we want to create a big web application which will have lot of features, admin related stuff, ecommerce website, we can have a product as a module, shipping as a module and these can be considered as different apps. All these apps in the website will be provided in one particular project which is your main website in general. Think of it like login stuff is in one app, showing the products is in one app.

How do we create an application named ***‘calc’*** ? – Let’s go to terminal in VSCode and activate the virtual environment that we have created earlier using ***workon test*** command and then observe that test is activated as highlighted below.

A screenshot of a computer

Description automatically generated with medium confidence

* Next command is to start the app calc



* After running the above command we can see that ***calc*** is created along side the previous telusko and we can see that the files are different in both the apps.

A screenshot of a computer

Description automatically generated with medium confidence

Discuss the files in general.

**First app in Django part 2 (Let’s make the app work by printing hello world on our webpage)**

Our task now is to make sure our home page shows hello world. To understand how things work we need to know how Django navigates stuff. The moment we request Django server or the moment you request the URL which is **locahost:8000** it will search of the URL’s mapping, what will happen is we are asking for URL homepage, so it will search for the mapping which is with the slash ( **/** ). So basically home page is referred as ‘/’ or empty. If you want to refer to a particular page website/documents(page for related documents) or website/products(page for products). So home page is usually referred to as empty or ‘/’.

* Here we have to do something with URL’s, this urls.py is for the entire project but not for the specific app, we want to work with specific app called calc. We need a file URLs.py inside the calc to have our own URL mapping.

Graphical user interface, text

Description automatically generated

* In order to create mapping let’s create a new file under calc with the name urls.py, observe the main urls.py file and grab

1. from Django.urls import path
2. From . import views (we need views as well)
3. Urlpatterns = we will mention the ‘ ’ or ‘/’ so we are calling the homepage, when the home page is called it will pick the function(views.home, name=’home’)(for mapping we will use a list of mappings, we might have multiple url’s to map with).
4. Make sure the code looks as shown below.

A screenshot of a computer

Description automatically generated with medium confidence

-🡪 save the urls.py file, VSCode might ask you to install an extension. Install it and proceed further.

There will be an error for views because the home function is not defined in the views.py as we are calling it from the views. So let’s define the function as shown below. It will be called home as we know, it will accept a request and it should return “hello world”. By doing so we are saying whenever a client **request** the home page we will return the hello world. Since client is requesting we need to send it to the client in a **response format**. Hence we use **HttpResponse.** Now there will be a problem with HttpResponse because whenever we use it we need to import it from a package **Django.http**

Text

Description automatically generated

Now if we go back to urls.py we should see the views.home error disappears. Will this work? – so this is how it looks like when the user calls the home page it will call the home function and the home function returns the httpresponse ‘hello world’

* If you run the runserver command and check the site we will still see the usual default page and NOT the hello world.

Till now we have done the changes to the files in **calc,** whereas the main project file is telusko in this case which controls everything, so we have to mention that entry in the main urls.py as well. Only one entry needs to be added which says something like this – ‘hey, you also have to use the urls.py file of the calc app’. We will use include function for this which needs to be imported from Django.urls as shown below

Text

Description automatically generated

* Now start runserver command and refresh/access the url which should show ‘hello world’ on home page.

If it doesn’t work troubleshoot the issue by finding the error

Steps recap:

1. When the request comes from the client it will go to the main urls.py as we are accessing the home page which speaks like this ‘ hey if you are searching for the home page you have to look for calc.urls’
2. Now we are in calc.ulrs, when home page is calling we will activate the home function within the views file.
3. The views file returns the ‘hello world’ as httpresponse

Text

Description automatically generated