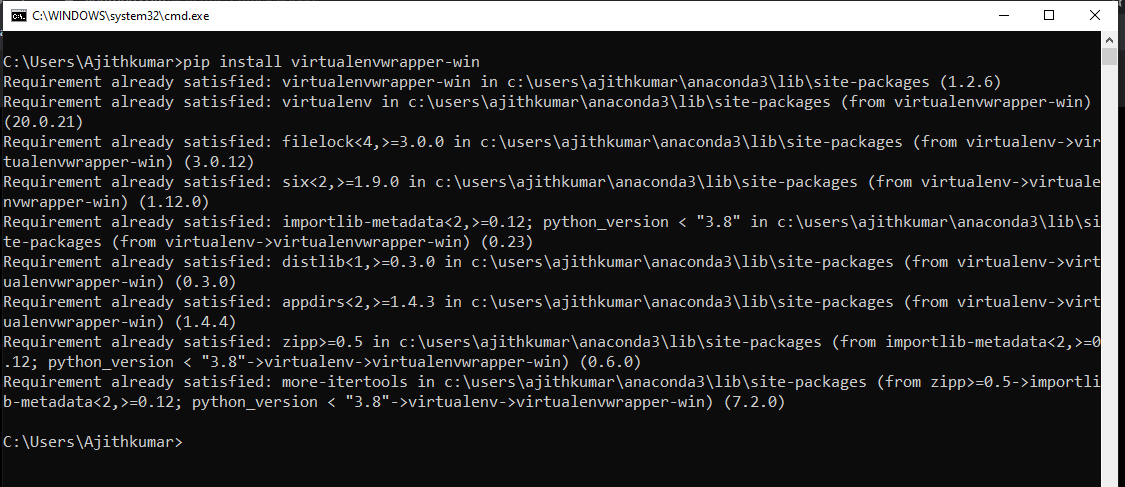
1. **Django Setup:**

We need to install different versions for each project, so we will install this in an virtual environment,

**Cmd :** ***pip install virtualenvwrapper-win***



***Cmd :******mkvirtualenv testaji***

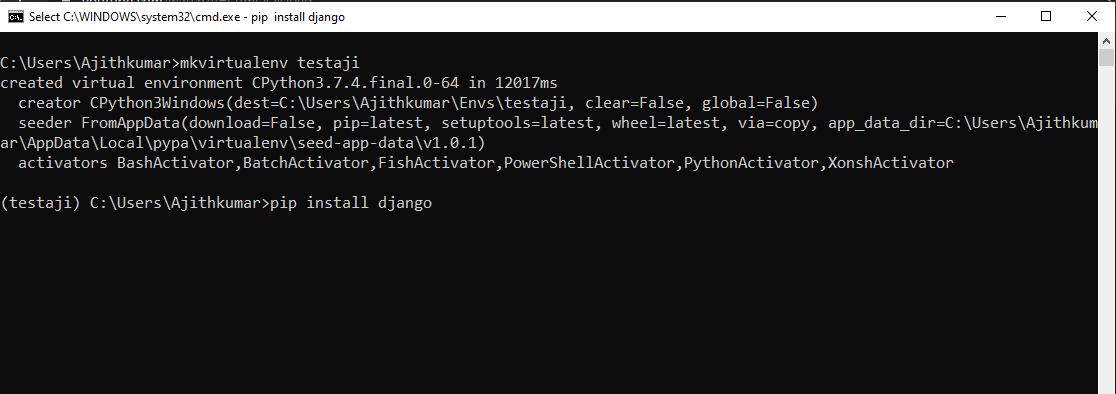
This command will create a new virtual directory.

Once you closed this command prompt and will join again mean, you need to use

**Cmd : *workon testaji***

**For installation cmd: *pip install django***

To check the version. Just use cmd***: django-admin –version***



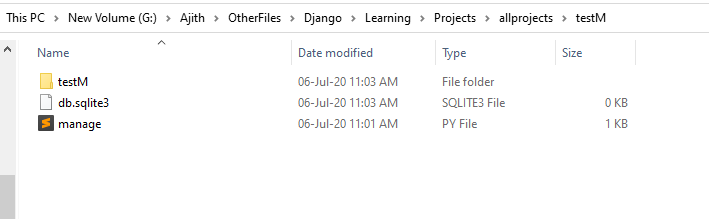
1. **Django Project creation :**

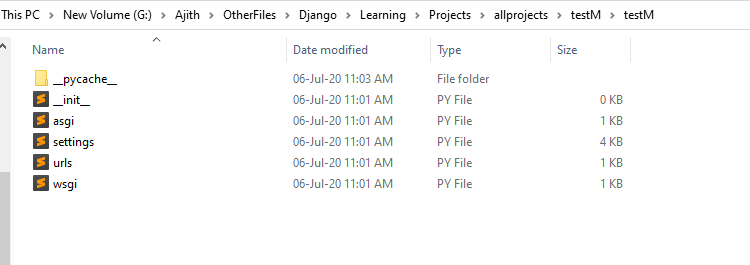
**Cmd :** ***mkdir allprojects***

**Cmd *: cd allprojects***

**Cmd*: django-admin startproject testM***

This will create a new project with few files in it as shown below.

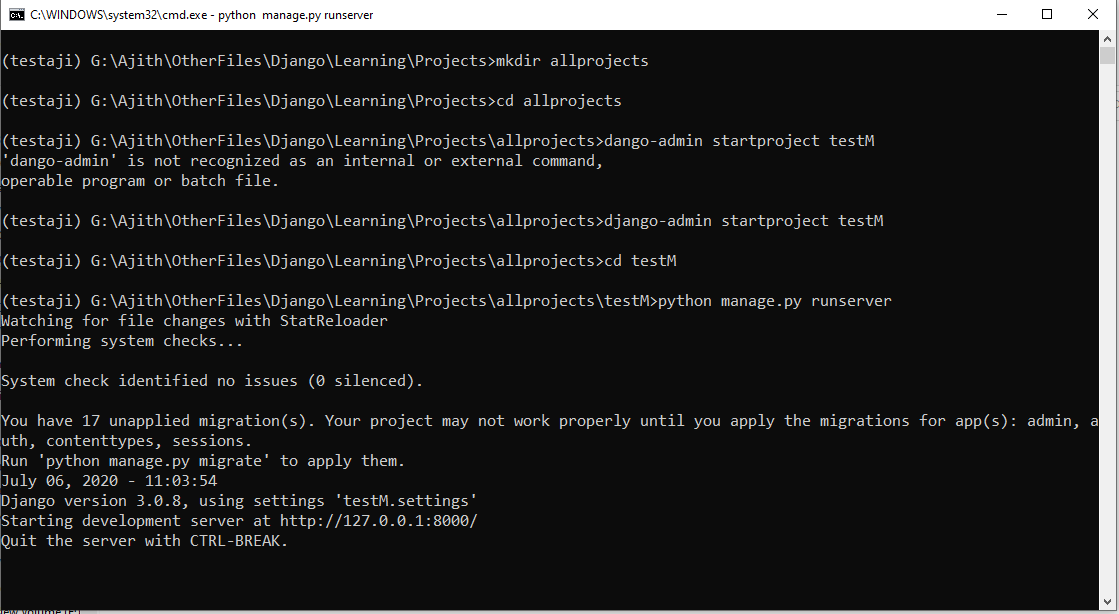




***Cmd : cd testM***

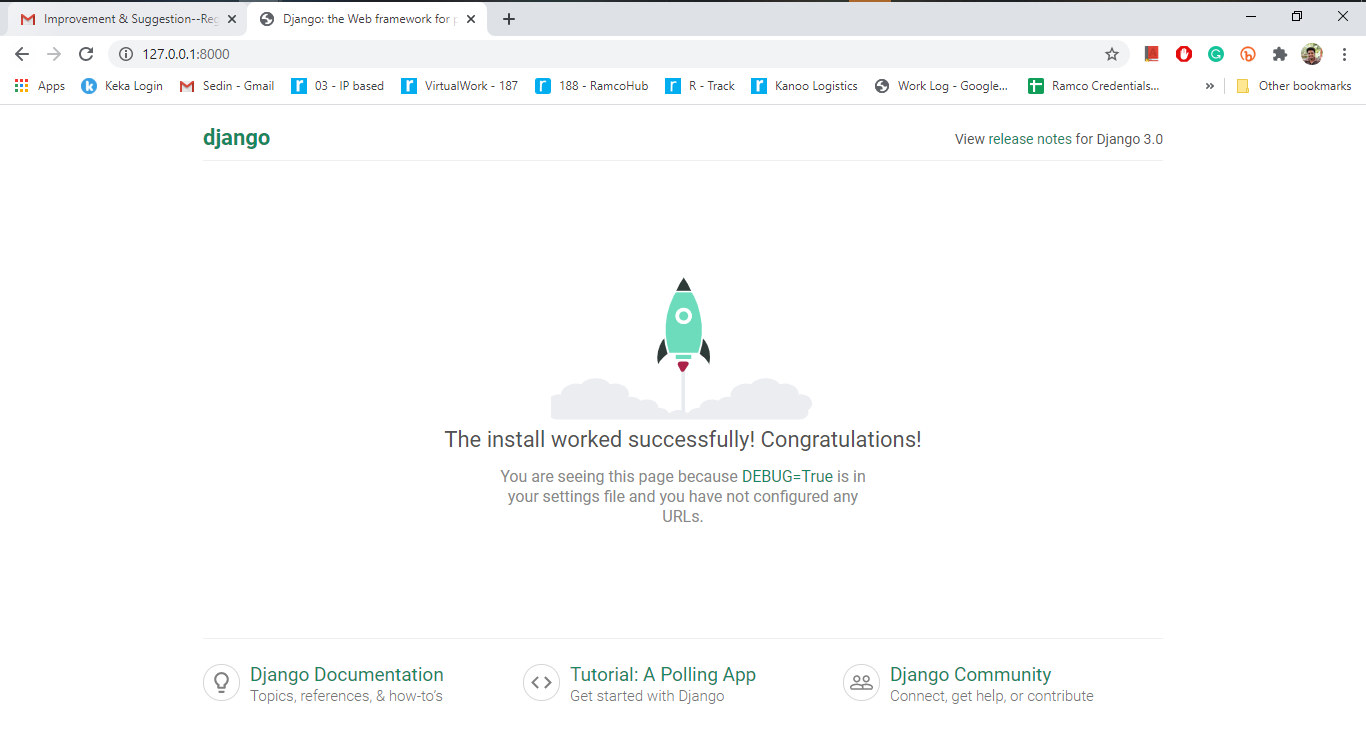
***Cmd : python manage.py runserver***

*Then the server will be started in some link as shown in the below image.*



***Localhost link :*** [***http://127.0.0.1:8000/***](http://127.0.0.1:8000/)

*This will open like the below link*



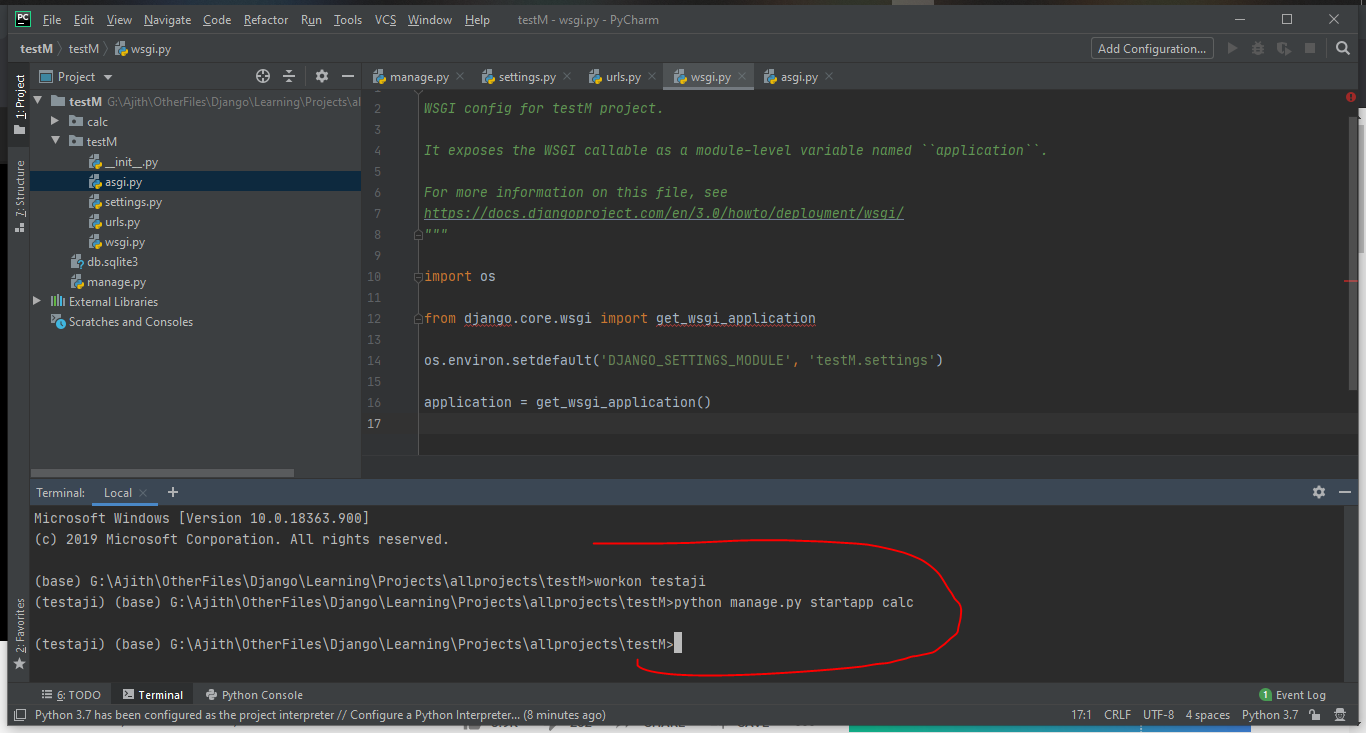
That’s all about a sample Project.

Now let create a sample application, a new app.

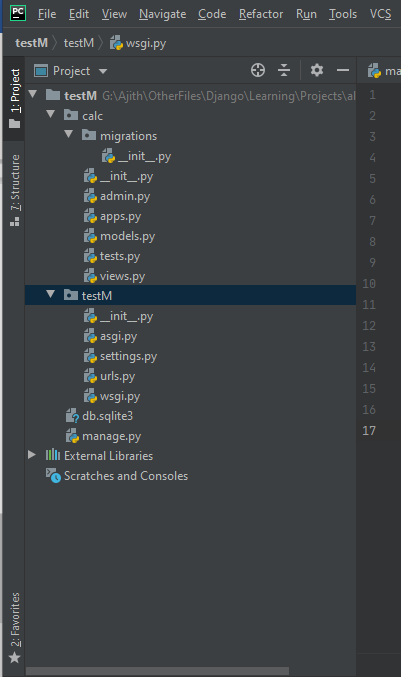
Open Pycharm editor and use the below command.

**Cmd : *workon testaji***

**Cmd: *python manage.py startapp calc***



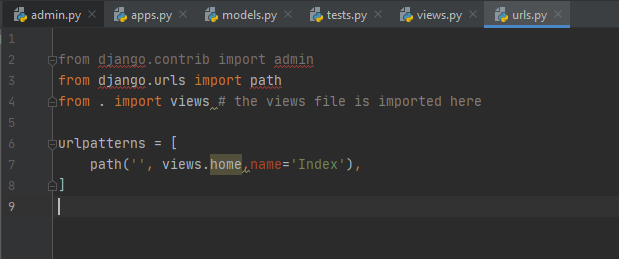
Once you run the command to create the application, then the below files will be created.



Just create a python file “urls.py ” and copy the piece of code from main urls.py

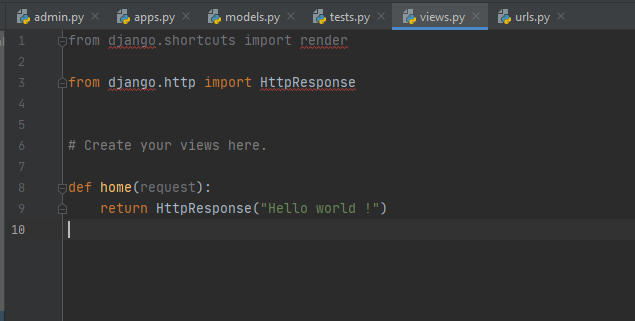
In urls.py ,

1. We need to mention and map the urls and the function to call. Usually the functions will be available in **views.py.**

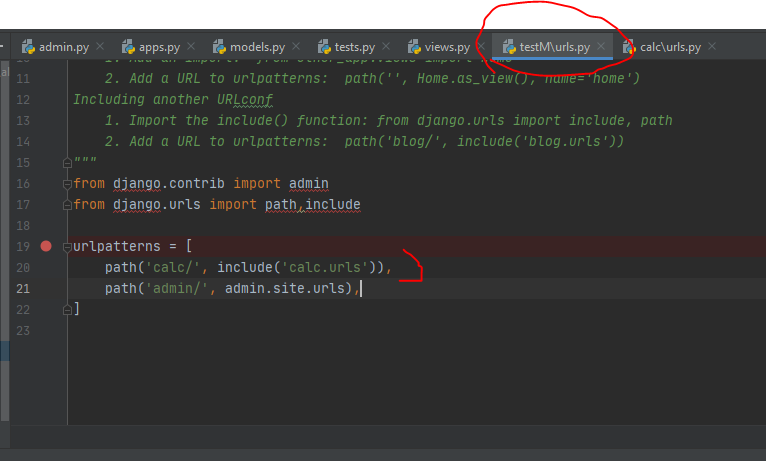


In views.py

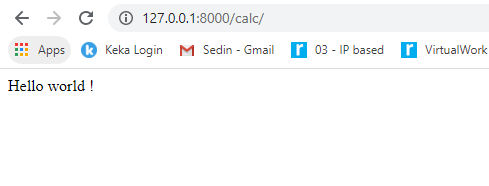
1. We need to write the function to return the response as ‘’Hello world”



Now we need to map these (calc) urls.py to the main urls.py (testM).

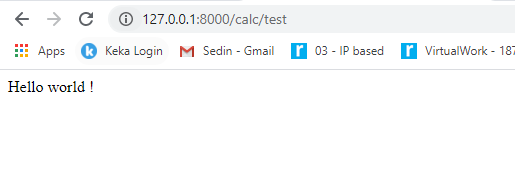


Then we need to check the same in the browser.

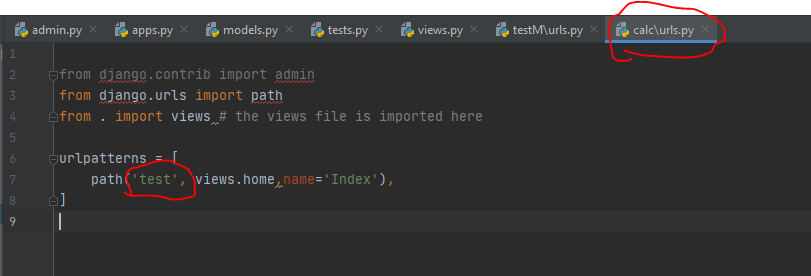


***Yes, Demo Project is success !!***

***A small hint ,***



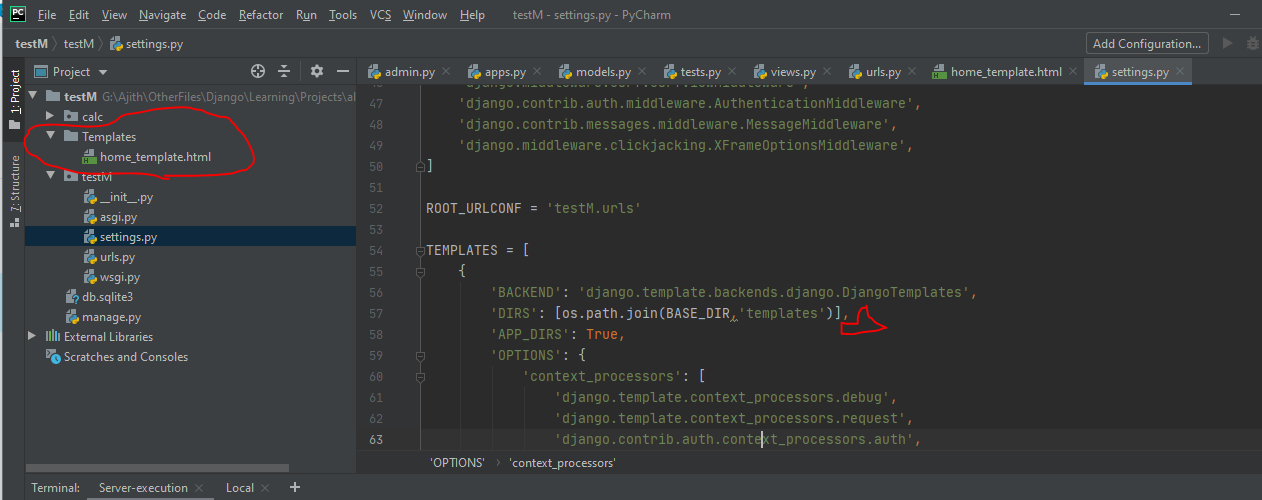
**Note :** the calc/test in the above image , as I have added a text in the urlpattern list of (calc - ) urls.py file .



If we just left it as it is, it will get executed. Otherwise it need a specific keyword.

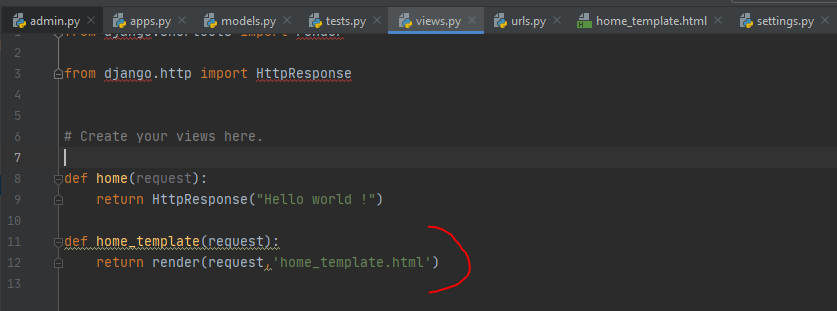
This is my own invention heh M. @?!

For Dynamic contents, we may use the “**Templates**” concept.

1. Create a folder named “**Template**” outside the calc app and map the path in (main) **settings.py .**
2. Then create a sample html template file.
3. \

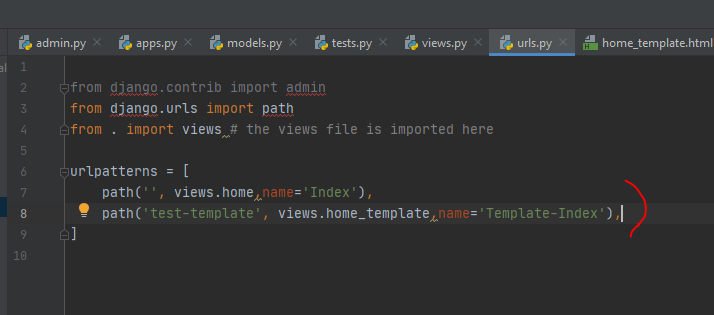
Create a new function in calc-view.py to return the template we just created in templates folder.

**Views.py**

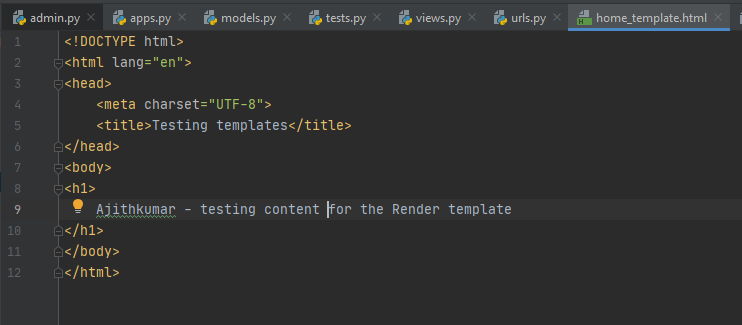


**Calc/urls.py :**

Here we are calling the respective function



**Html content :**



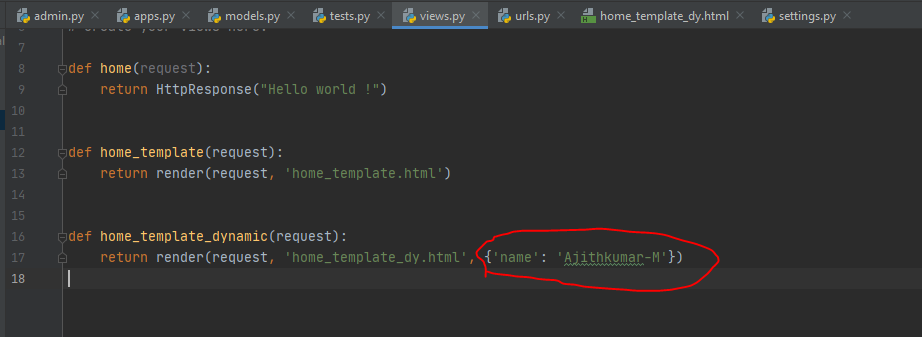
**Browser – testing Successfully.**



**Now we just created a html template and tested it by calling.**

**More dynamic content.**

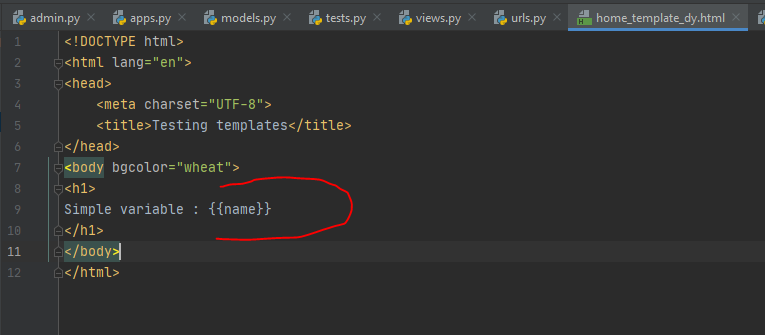
**Views.py**



**New template added and called the same in views.py.**

**And added a Jquery format in the new html file .**

**So this makes the run time data passage possible.**



**Output :**

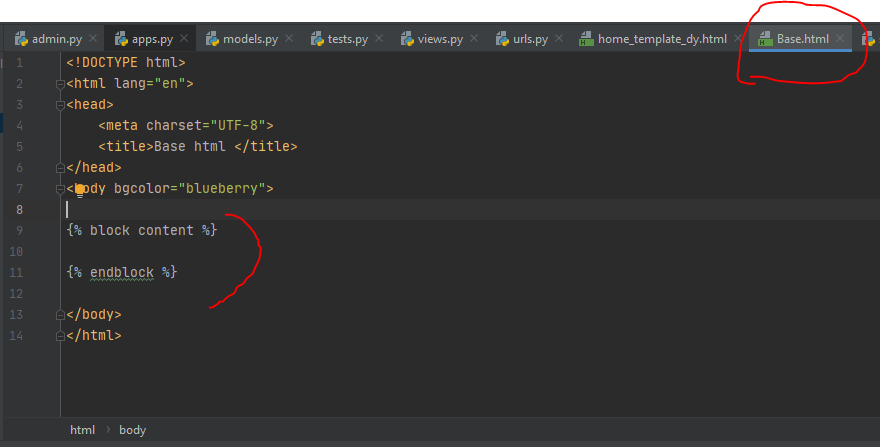
**08/07/2020**

How to add a HTML template into another HTML template.

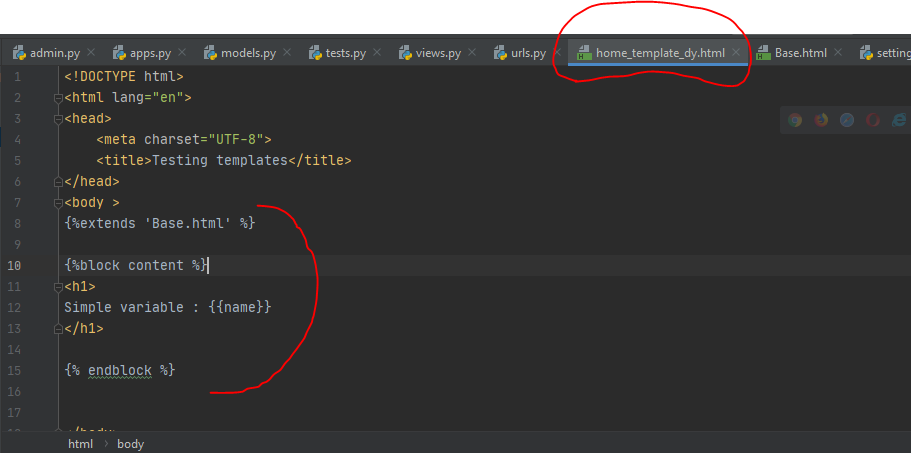
For that we need to have the “**Jinja template”**

**Base.html** is a new filewhich is created using Jinja template with the sample codes.

{% block content %}  
  
{% endblock %}



Then we need to call the above template in our templates.



Once we just passed a data into the html page and just printed it.

Now going to create a form which can add two numbers.

For this we are adding code in “home\_template”

**For Actual data passage and date processing.**

From a html page we need to give the form action.

1, using action=’add’ ( this add is a url , so we need to add this to urls.py.

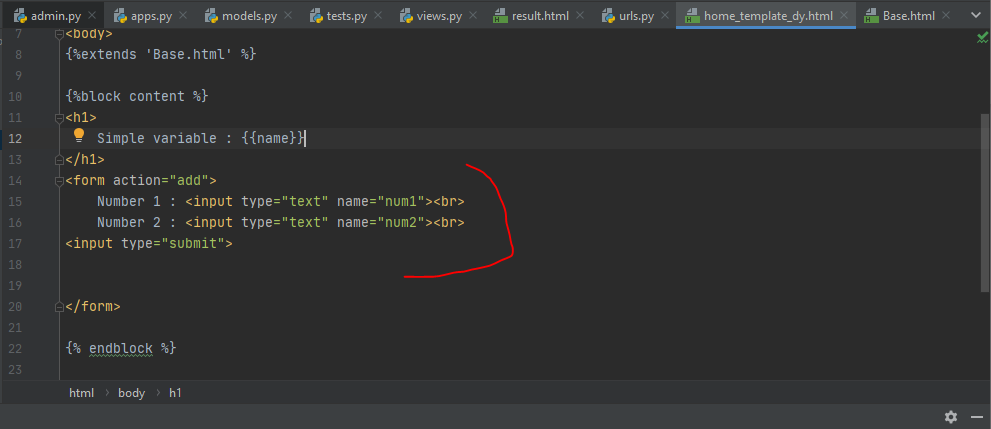
2, Once we have added the add page and the function name ,then we need to add these things into **views.py.**

3, In the view. Py we can add these values / perform whatever operations we want to perform and return the data to a html page.

4, using asusual render template function in “views.py”

***Below images will demonstrate these things.***

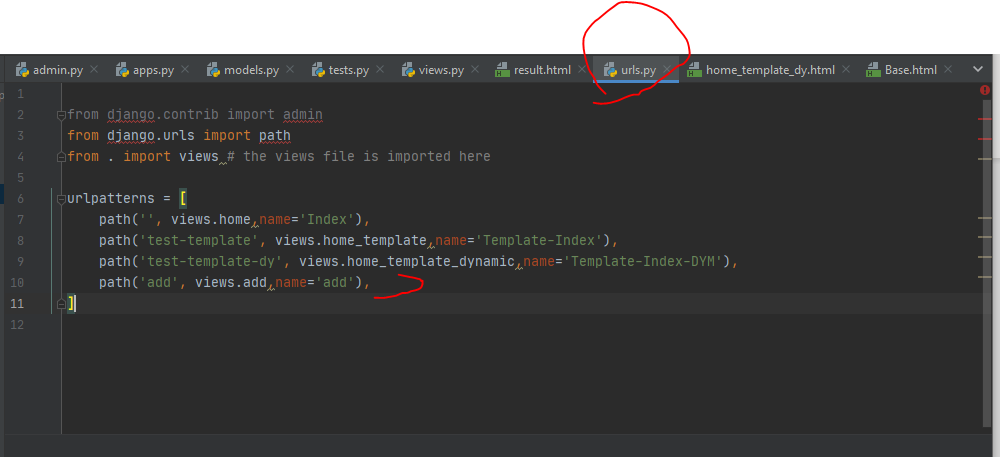
An initial home page “**test-template-dy.html**”

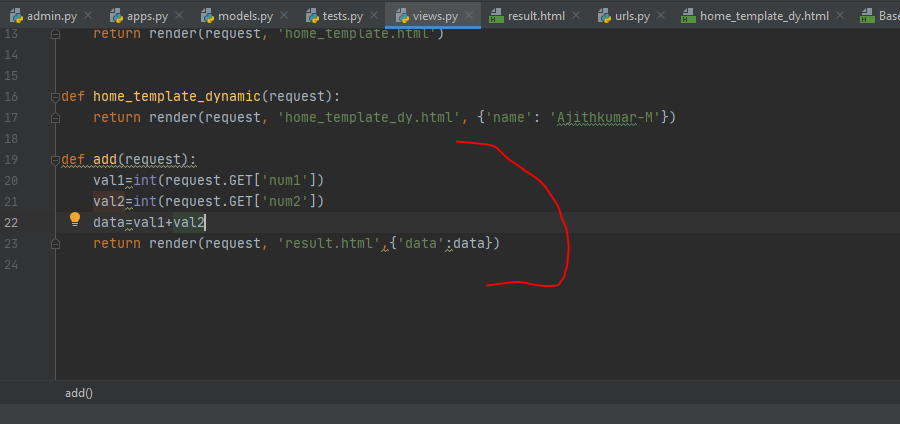


Once we add these mentioned lines , we can check these things on the web page we developed.

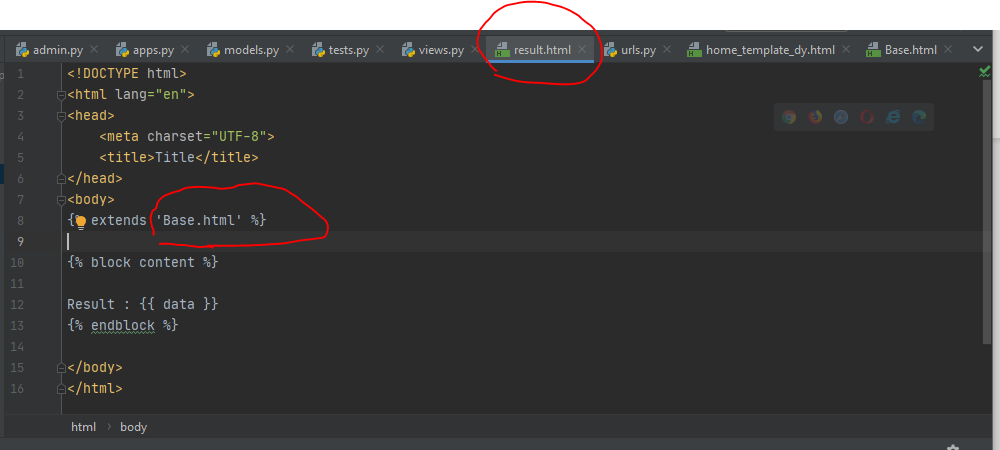
Then we need to add these action into the **urls.py** and then need to write the definition for these function ‘add’ in **views.py**





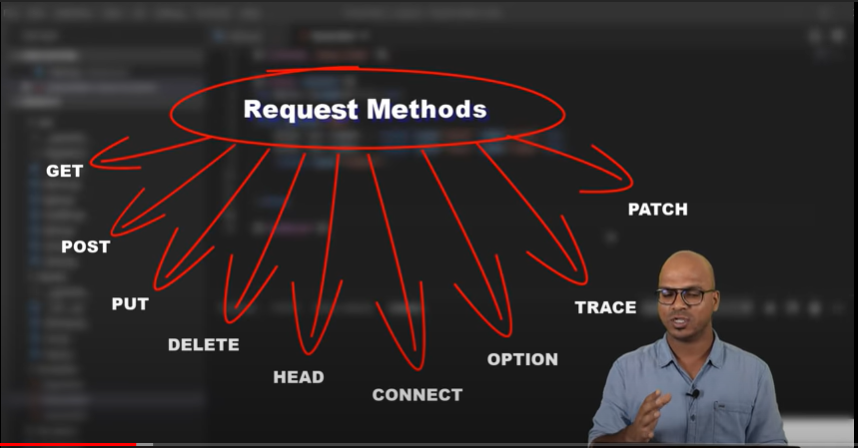


New result template “**result.html**”



That’s all about the basic view of an web application.

Let’s see about the request in more detail.



GET – basically fetching the data from the server.

POST- sending data to the server.