Download Python 3.x

Download Anaconda

Download and install Pycharm

Now on Pycharm:

Create env:

conda create -–name env\_name djanjo=2

Deactivate default env:

deactivate

Activate Django env:

activate env\_name

To create Project:

django-admin startproject project\_name

Enter to project directory:

cd project\_name

Verify by:

python manage.py runserver

Create App:

python manage.py startapp app\_name

Create views in app\_name/views.py:

from django.shortcuts import render  
from django.http import HttpResponse  
  
def index(request):  
 return HttpResponse("Hello there, I got something !!")

Now Url Config:

Right click on app\_name -> new -> python file (give file name Urls)

On new file app\_name/Urls.py

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('',views.index, name='index')  
]

Now point root Url to our app url file:

By updating project\_name/Urls.py:

from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', include('app\_name.urls'))  
]

Now check the progress by:

<http://127.0.0.1:8000/app_name>

**Template\_Tagging:**

Create New directory in root of project [in /project\_name in this case] and name that directory ’template’

Create new html file inside template directory and type something inside body tag to ensure the view:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Django Template</title>  
</head>  
<body>  
<h1>Hello From index.html !!</h1>  
</body>  
</html>

Now we have to configure the template directory on setting. Py

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
TEMPLATE\_DIR = os.path.join(BASE\_DIR,'template')

And include the template dir just created to list of directories in same file

'DIRS': [TEMPLATE\_DIR,],

Now we have created view to render the template we created in views.py.

def temp\_demo(request):  
 dist\_var ={'any\_key': "hello, this is a value for the key from view.py that can be used in index.html"}  
 return render(request,'index.html', context=dist\_var)

Let’s include that key just created in out index.html

<body>  
<h1>Hello From index.html !!</h1>  
{{ any\_key }}  
</body>

Finally, we need to update the urls.py for path of template, pass the default to template and we can keep first with app\_name

urlpatterns = [  
 path('tempapp',views.index, name='index'),  
 path('',views.temp\_demo, name='index1')  
]

Now lets run the server it its not live already. We should get the something like



**STATIC Files**

It’s just similar to template tagging, Lets start by creating directory named static in root of project, then other two directory named images and css inside the static folder. We keep static files like images and css inside ststic directory.

Paste any sample image of our choice inside image directory.

And Similar to template lets update setting.py for the static directory

TEMPLATE\_DIR = os.path.join(BASE\_DIR,'template')  
STATIC\_DIR = os.path.join(BASE\_DIR,'static')

And

'BACKEND': 'django.template.backends.django.DjangoTemplates',  
'DIRS': [TEMPLATE\_DIR,STATIC\_DIR,],

Then, simply load the static image file in index.html

Right below the doctype tag:

{% load staticfiles %}

And anywhere you like inside body tag:

<img src= "{% static "images/sample.jpg" %}" alt="This will appear if no image found">

Point to 127.0.0.1.:8000 to check if everything is working.

CSS

It’s a simple styling for out html template, we had already created css directory, lets now create the file called style.css inside it so we can write some css script on it

h1{  
color: red;  
text-align: center;  
}

Then we have to link the file inside head tag in our index.html as:

<link rel="stylesheet" href="{% static "css/style.css" %}">

Then simply check the difference by refreshing 127.0.0.1.:8000 page.

**HTML MODELS**

Models are the place where we create the table for the database by creating class for each tables. Basically each class are the tables that will be created.

Lets create three tables Employee, Hours and Salary in models.py

from django.db import models  
  
class Employee(models.Model):  
 emp\_name=models.CharField(max\_length=200,unique=True)  
  
 def \_\_str\_\_(self):  
 return self.emp\_name  
  
class Hours(models.Model):  
 emp\_name = models.ForeignKey(Employee, on\_delete=models.PROTECT)  
 days = models.DateTimeField('date')  
 hours = models.IntegerField  
  
 def \_\_int\_\_(self):  
 return self.days  
  
class Salary(models.Model):  
 days = models.ForeignKey(Hours, on\_delete=models.PROTECT)  
 Credeted\_salary = models.IntegerField()  
 salary\_date = models.DateField()

then we need to migrate the changes to the database that can done by command

python manage.py migrate

Django already have built in admin panel and we just need to create super user to access first.

Python manage.py createsuperuser

Provide username, email and password

Now run the server again

Python manage.py runserver

And point to <http://127.0.0.1:8000/admin> and login using the credential created above.

Now we need to update in admin.py

from django.contrib import admin  
from tempapp.models import Employee,Hours,Salary  
# Register your models here.  
admin.site.register(Employee)  
admin.site.register(Hours)  
admin.site.register(Salary)

Now we need to migrate the database and tables

python manage.py makemigrations

python manage.py runserver