Django Project Guide

**Steps for Project setup:-**

1. First create a virtual environment – ( cmd- virtualenv space your environment name)
2. Then activate of your environment using – your environment name\Scripts\activate.
3. Once activate your environment then run this command – pip install django
4. Once django install your system then run this command – django-admin startproject projectname .
5. cd projectname(project root)
6. Then install first app using this command – django-admin startapp app\_name or python manage.py startapp appname
7. Then install the app in here-

INSTALLED\_APPS = [  
 **'django.contrib.admin'**,  
 **'django.contrib.auth'**,  
 **'django.contrib.contenttypes'**,  
 **'django.contrib.sessions'**,  
 **'django.contrib.messages'**,  
 **'django.contrib.staticfiles'**,  
 **'myApp'**,# You have to provide your app name here  
]

1. Create a templates folder in your project root
2. Settings.py:

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR,'templates'),], # push your template folder name

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

1. Then run the server using this command – python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

September 10, 2020 - 09:16:24

Django version 3.1.1, using settings 'ecommerce.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

1. Click this <http://127.0.0.1:8000/> link . it will open in web browser.
2. To print ‘hello world’ you have to do – goto to views.py file ------🡪

* def FunctionName(request):

return render(request,’index.html’)

* index.html->

<html>

<head>

<title>My Webpage</title>

</head>

<body>

<h1>Hello world</h1>

</body>

</html>

* urls.py

from appname.views import functionname

path(‘’,functionname),

* Then run the server using pyton manage.py runserver
* Database Connection

Steps:

1. Models.py

# here create your own model

from django.db import models

class ModelName(models.Model):

#declare your models fields

Name= models.CharField(“name”,max\_length=255)

Email = models.EmailField(“email”,max\_length=100)

Dob = models.DateTimeField(auto\_add\_now=True)

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

return self.name

1. Views.py

from .models import ModelName

def model\_view(request):

name = ModelName(

Name = “sovan”,

Email = [sovan.pbc@gmail.com](mailto:sovan.pbc@gmail.com)

Name.save()

Obj = ModelName.objects.all()

return render(request,”db\_view.html”,{“Obj”:Obj})

1. Urls.py

from appname.views import model\_view # import your view function

path(“db”, model\_view),

1. db\_view.html # in your templates folder create a html file name as db\_view.html

<h1>Db records </h1>

{{ Obj }}

# web browser return a query set

Using python shell we can push records

Steps-

Cmd- python manage.py shell

>>> from appname.models import ModelName

Obj = ModelName(

Name= “sovan”,

Email = [sovan.pbc@gmail.com](mailto:sovan.pbc@gmail.com)

)

Obj.save()

Cmd for show your records --🡪 ModelName.objects.all() -> it returns a query set