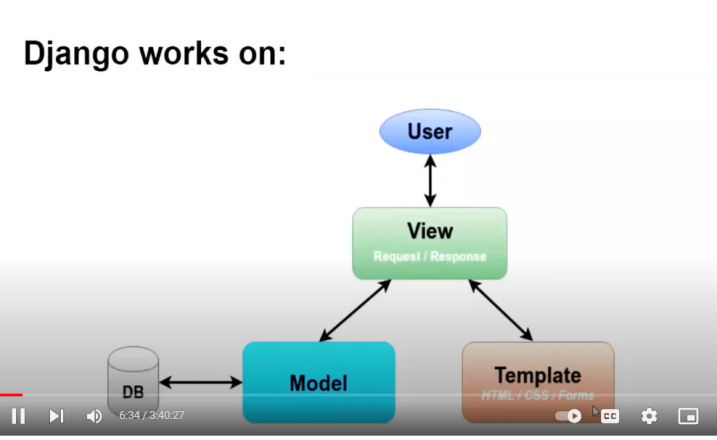


Views will take the responses

Django it is a frame work(to built the appilications)



MVT (model view Template) using this architecture it will work

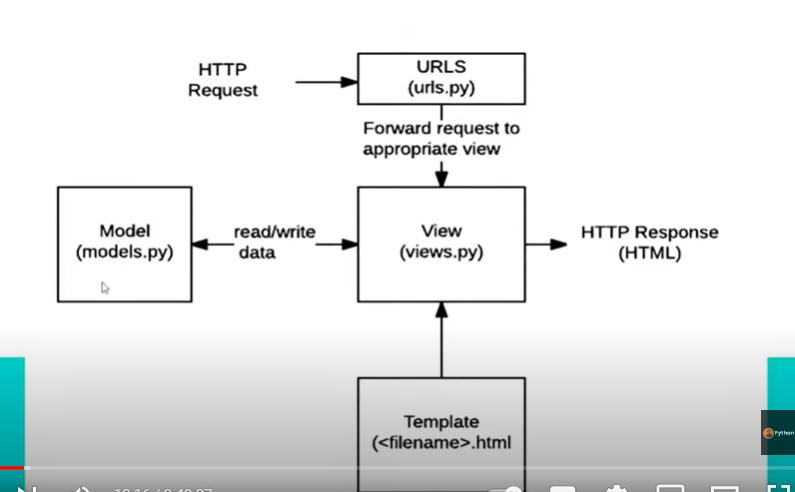
User and view has url link

View (request / response) - Template(HTML/CSS/FORMS)

ORM - object Relational Mapping(free defined tables )

MODEL VIEW TEMPLATE

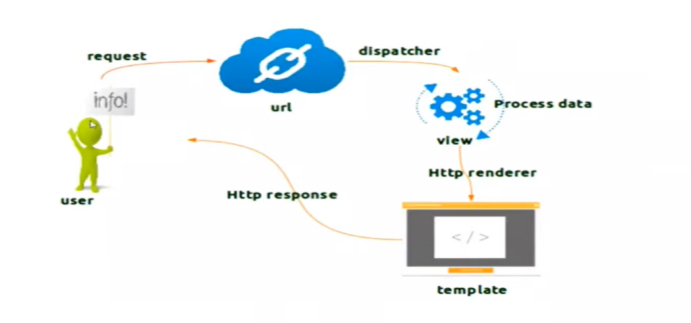
VIEW WILL GET REQUEST (URL) MODEL CONTAINS THE DATA (READ/WRITE DATA)



**Model** - It is responsible for maintaining the data

**View** - it acts as a link between model data and the template

**Template** - consists of static parts of the desired html o/p as well as some special syntax describing how dynamic content will be inserted.



**DJANGO SETUP**

1. **Pip install virtualenv**
2. **Virtualenv filename. Or instead of this w can use python -m venv kranthi**
3. **Open filename open scripts**
4. **Activate**
5. **Django-admin startproject project name**

Views are two types : 1) function based

1. class based

Views.py :

from django.shortcuts import render

from django.http import HttpResponse

def home(request):

    return HttpResponse("thisssss is kk")

def demo(request):

    return HttpResponse("hii guys")

Url.py

from django.contrib import admin

from django.urls import path

from .views import home, demo

urlpatterns = [

    path('admin/', admin.site.urls),

    path("", home, name="home"),  # Home view for the root URL

    path("demo/", demo, name="demo"),  # Demo view for the "/demo/" URL

]

**render** function is used to combine a template (HTML file) with data (context) and return an HTTP response that contains the generated HTML content. Essentially, render takes a template, fills it with data, and returns a complete HTML page as an HTTP response.

**Settings.py :**

1. In installed apps configure the ‘demoapp’,
2. In templates in dirs (take from the github) in os we are creating or it was in templates
3. New folder and create templates
4. Data base default as sqllite or if we don’t like means then then take it from github
5. If we want paste the static(css and js) files also
6. Create folder static
7. Media confi(images videos ….) if we want paste it…
8. After that go to urls in project (demo1) and insert
   1. From django.conf.urls.static import static
   2. From djangoconf import settings
9. And copy the urlpatterns += from github

**STATIC Files**

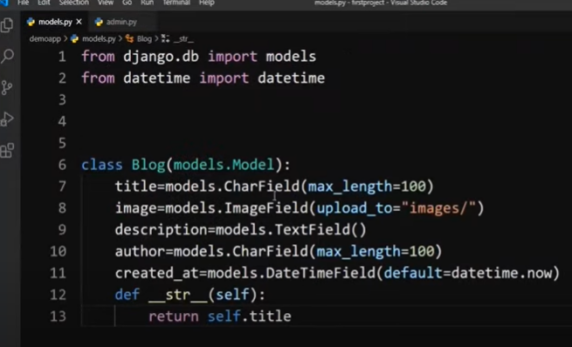
**All from bootstrap : -----**

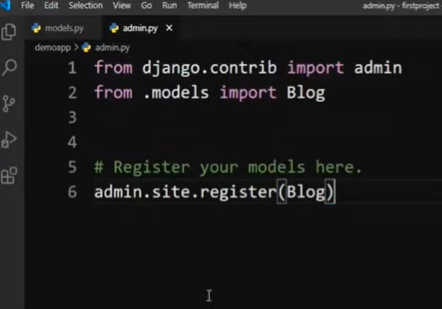
**Templates (html pages) :**

1. In template create a file name called base.html
2. Now connect in view (return render(request,”base.html”)
3. If we want to give name by user then in view (return render(request,”base.html”,{“name”:”kk”})
4. And also change it in base.html also as <h1> my name is {{name}} </h1>

**MODELS>PY:** it contains the essential fileds and behaviors pf the data you are storing

Create user : python manage.py create superuserthen it will ask user name and password and email address also

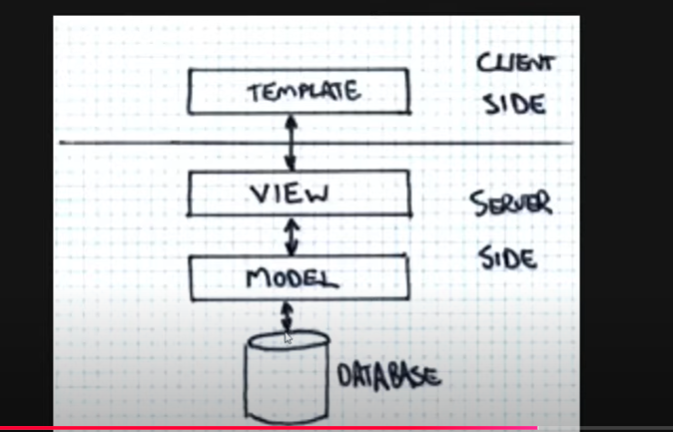
1. Models will be in apps only…
2. 
3. **Now confi it in admin**

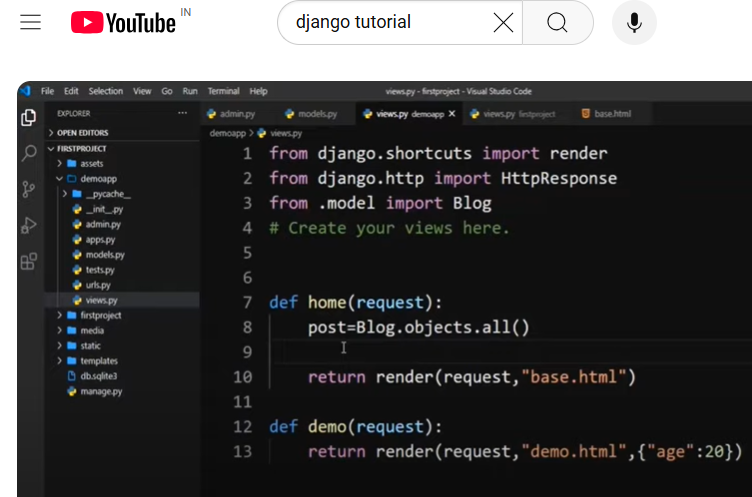


Python manage.py makemigrations

Python manage.py migrate

**DATA BASE TO Template:**





Post is a variable - and in sql we use select \* from table table\_name but in this we use blog.objects.all()

<!DOCTYPE html>

<html>

<head>

    <title>Page Title</title>

</head>

<body>

    <style>

        h1 {color : green}

    </style>

    <h1>This is a Heading</h1>

    <p>This is a paragraph.</p>

</body>

</html>