Django is open source web development framework.

-> It is easy to learn and fast to develop.

-> It follow the MVC module.

MOdule view Controller (MVC)

Advantage :

1. Fast

2. Component

3. Security

4. Scalablity

--------------------------------

install virtual environment ;

-> pip install virtualenvwrapper-win

Create virtual len environment

-> mkvirtualenv test

Install django

-> pip install django

Get to know version fo django

-> django-admin --version

Create seperate folder and create project inside it.

-> django-admin startproject <project\_name>

Above command will create project and verify there should be manage.py file exist.

and inside the project folder name

filename Exist

--> urls.py : it will contains all the urls of project

--> wsgi.py : it will contains information to host project locally.

-------------------------------create super user in django-----------------

$ python manage.py createsuperuser

Enter your desired username and press enter.

Username: admin

You will then be prompted for your desired email address:

Email address: admin@example.com

The final step is to enter your password. You will be asked to enter your password twice, the second time as a confirmation of the first.

Password: \*\*\*\*\*\*\*\*\*\*

Password (again): \*\*\*\*\*\*\*\*\*

Superuser created successfully.

-------------------------------------------

# Run the Server using below command :

-> python manage.py runserver

#Once above command will run, light wait server will start like.

#You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

July 24, 2019 - 14:59:48

Django version 2.2.3, using settings 'LearningApp.settings'

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

# Copy paste the this url to browser and verify the django should be launched sucessfully.

manage.py : A command-line utility that lets you interact with this Django project in various ways. You can read all the details about manage.py in django-admin and manage.py.

The inner mysite/ directory is the actual Python package for your project. Its name is the Python package name you’ll need to use to import anything inside it (e.g. mysite.urls).

mysite/\_\_init\_\_.py : An empty file that tells Python that this directory should be considered a Python package. If you’re a Python beginner, read more about packages in the official Python docs.

mysite/settings.py : Settings/configuration for this Django project. Django settings will tell you all about how settings work.

mysite/urls.py : The URL declarations for this Django project; a “table of contents” of your Django-powered site. You can read more about URLs in URL dispatcher.

mysite/wsgi.py : An entry-point for WSGI-compatible web servers to serve your project. See How to deploy with WSGI for more details.

# Now Lets ahead and make changes in project :

mysite/settings.py -> Debug = True : It will give you lot of information, whats going on inside the project

# When you hit url on browser , then django search for URL or anyother changes in continuos manner.

# Now Lets create app using command

-> python manage.py startapp <appname>

-> example : python manage.py startapp calc

# We have to create separate url file for app, we wont change into project url file.

1. Now do some mapping in app urls.py

from django import urls

from . import views

from django.urls import path

urlpatterns = [

path('', views.home, name='home')

]

2. In views.py defind home method, which will return hello world along with Htttpresponse.

from django.http import HttpResponse

# Create your views here.

def home(request):

return HttpResponse("Hello World")

3. Now do app urls mapping in project urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('',include('calc.urls')),

path('admin/', admin.site.urls),

]

4. Now run the server using command : python manage.py runserver

and verify hello world msg should be displayed on browser.

EVERY THINGS WORKS FINE : Now LETS MOVE AHEAD TOMWARD TEMPLATE :

------------------------TEMPLATE --------------------------------

5. HTML CONTENT : We can defined html content in home method in views.py file of app and we can pass as response.

from django.http import HttpResponse

# Create your views here.

def home(request):

return HttpResponse("<h1> Hello World </h1>")

save it and it will reflect on home page

6. DTL : Django Template Language : user have to create templates directory in root directory

where all the html file will be exist.

-> Create templates directory in root directory of project.

-> For template perpouse we have to some changes in settings.py

Base content for setting template without any changes :

TEMPLATES = [

{

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

'DIRS': [],

'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',

],

},

},

]

Content for setting with any changes :

TEMPLATES = [

{

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

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

'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',

],

},

},

]

-> Inside the template create home.html file and set contents with tag.

<h1> Hello World </h1>

-> Create method in calc/view.py with name html\_home and render the template.

from django.shortcuts import render

from django.http import HttpResponse

def html\_home(request):

return render(request, 'index.html')

-> Now map this method name in calc/urls.py

from django import urls

from . import views

from django.urls import path

urlpatterns = [

path('', views.home, name='home'),

path('index/', views.html\_home, name='html\_home'),

]

-> Lets refresh browser and look to https://0.0.0.127:8000/index then home.html page should

be render and html page data should be available.

-> If you any multikey maping issue, then restart the browser and start again.

7. Now update the dynamic data on html page.

-> Create dymanic\_home.html file in templates directory and set contents with tag.

<h1> Hello {{name}} </h1> # this is called jinja template.

-> Create method in calc/view.py with name html\_home and render the template.

from django.shortcuts import render

from django.http import HttpResponse

def dhtml\_home(request):

return render(request, 'dynamic\_home.html', {'name':'Deepesh'})

-> Now map this method name in calc/urls.py

from django import urls

from . import views

from django.urls import path

urlpatterns = [

path('', views.home, name='home'),

path('index/', views.html\_home, name='html\_home'),

path('dindex/', views.dhtml\_home, name='dhtml\_home'),

]

-> Lets refresh browser and look to https://0.0.0.127:8000/index then home.html page should

be render and html page data should be available.

Should print : Hello Deepesh

---------------------------Use of base.html --------------

8. base.html use for same for all the html pages on project.

-> Create base.html file in template directory

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewpoint" content="width=device-width, initial-scale=1.0">

<title>ITPD</title>

</head>

<body bgcolor="cyan">

{% block content %}

{% endblock %}

</body>

</html>

-> create another html file in template directory, which will extent this base.html file

#dhtml\_with\_base.html

{% extends 'base.html'%}

{% block content %}

<h1>Hello {{name}} </h1>

{% endblock %}

-> do the changes in calc/urls.py and calc/views.py

#urls.py

from django import urls

from . import views

from django.urls import path

urlpatterns = [

path('', views.home, name='home'),

path('index/', views.html\_home, name='html\_home'),

path('dindex/', views.dynamic\_home, name='dhtml\_home'),

path('base/', views.dynamic\_home\_base, name='dhtml\_home\_base')

]

#views.py

from django.shortcuts import render

from django.http import HttpResponse

# Create your views here.

def home(request):

return HttpResponse("<h1>Hello World</h1>")

def html\_home(request):

return render(request, 'index.html')

def dynamic\_home(request):

return render(request, 'index\_dynamic.html', {'name':'Deepesh'})

# This method will render index\_with\_base.html with base.html extention.

def dynamic\_home\_base(request):

return render(request, 'index\_with\_base.html', {'name':'Rahul'})

------------------------------ GET DATA FROM FORM AND PERFORM OPERATION ON IT --------------------

8. Read data from from and perform some operation.

-> Create html file with form content: addition.html which content form and two text build

and as a action it will call add method.

# addition.html

{% extends 'base.html'%}

{% block content %}

<h1>Hello, {{name}} </h1>

<form action="add">

Enter 1st Number :<input type="text", name="num1"><br>

Enter 2st Number :<input type="text", name="num2"><br>

<input type="submit">

</form>

{% endblock %}

-> Create another html file which will show the result on web page.

#result.html

{% extends 'base.html'%}

{% block content %}

<h1>Result : {{result}} </h1>

{% endblock %}

-> Add function and addition and add in view.py, and rerun the server.

def addition(request):

return render(request, 'addition.html', {'name':'Rahul'})

def add(request):

num1 = request.GET['num1']

num2 = request.GET['num2']

result = int(num1) + int(num2)

return render(request, 'result.html', {'result':result})

-------------------------------------- GET and POST METHOD ----------------------------------------------------

1. GET : When we use get method , then data we are sending will be visible on address bar

Generally get method to get data from server.

2. POST : When we user post method, then data we are sending will be submit to the server.

-> Previously we worked on get method now let’s work with post method.

-> Create two html files one will accept users details and another will display the user detail.

with post post.

-> When we submit data , if we want to protect from hackers thne we will use csrf token to protect the data:

CSRF : Cross-site request forgery, also known as one-click attack or session riding and

abbreviated as CSRF (sometimes pronounced sea-surf) or XSRF, is a type of malicious exploit of

a website where unauthorized commands are transmitted from a user that the web application trusts

**getdata\_post.html**:

{% extends 'base.html' %}

{% block content %}

<form action="display" method="post">

{% csrf\_token %}

# CSRF token help user avoid attach from hackers.

Firstname <input type="text", name="fname"></br>

Lastname <input type="text", name="lname"></br>

Email ID <input type="text", name="email"></br>

Address <input type="text", name="address"></br>

Mobile No <input type="text", name="mobile"></br>

<input type="submit">

</form>

{% endblock %}

**display\_post.html**:

{% extends 'base.html' %}

{% block content %}

<table align="Center" border="1" bgcolor="yellow">

<th colspan="2">User Details</th>

<tr>

<td>Firstname</td>

<td>{{first\_name}}</td>

</tr>

<tr>

<td>Lastname</td>

<td>{{last\_name}}</td>

</tr>

<tr>

<td>Email ID</td>

<td>{{email\_id}}</td>

</tr>

<tr>

<td>Address</td>

<td>{{address}}</td>

</tr>

<tr>

<td>Mobile Number</td>

<td>{{mobile\_no}}</td>

</tr>

</table>

{% endblock %}

-> Make changes in view.py and urls.py in calc directory.

**views.py**:

from django.shortcuts import render

from django.http import HttpResponse

def addition(request):

return render(request, 'addition.html')

def result(request):

number1 = request.GET['num1']

number2 = request.GET['num2']

resultdata = int(number1) + int(number2)

return render(request, 'result.html', { 'result': resultdata })

def get\_user\_data(request):

return render(request, 'getdata\_post.html')

def display\_user\_data(request):

first\_name = request.POST['fname']

last\_name = request.POST['lname']

email\_id = request.POST['email']

address = request.POST['address']

mobile\_no = request.POST['mobile']

user\_data = {}

user\_data['first\_name'] = first\_name

user\_data['last\_name'] = last\_name

user\_data['email\_id'] = email\_id

user\_data['address'] = address

user\_data['mobile\_no'] = mobile\_no

return render(request, 'display\_post.html', user\_data)

urls.py:

from django import urls

from . import views

from django.urls import path

urlpatterns = [

path('addition/', views.addition, name='addition'),

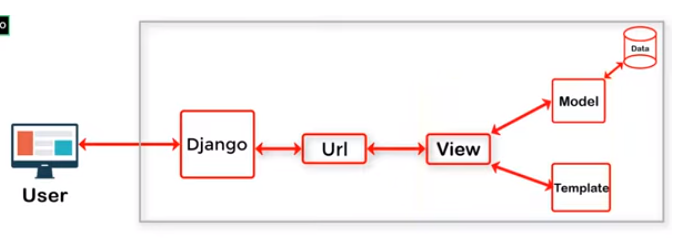
path('addition/result', views.result, name='result'),

path('userdata/', views.get\_user\_data, name='userdata'),

path('userdata/display', views.display\_user\_data, name='displaydata'),

]

**MVT (Module View Template)**



**Model** : It will contains all the database related connection and object

**Template**: It will contains all the web page for input output of data.

**View:** It will contains all the business logics behind and any website.

**Urls** : It contains all the mapping and data flow user mapping from the users.

**WORK WITH STATIC TEMPLATES**

* **Download static website from given URL:** <https://colorlib.com/wp/template/travello/>
* Till then we have worked inside the calc app.
* Now we will create new app with name travelo.
* Inside root directory use this command :
* python manage.py startapp travelo.
* Now unzip the template zip file and copy the index.html file into project template folder.
* Create separate urls.py file in side travel app and do mapping with view.py file of travel.

travelo/view.py

**from** django.shortcuts **import** render  
  
*# Create your views here.***def** travelo\_home(request):  
 **return** render(request, **'index.html'**)

travelo/urls.py

**from** django **import** urls  
**from** . **import** views  
**from** django.urls **import** path  
  
  
urlpatterns = [  
 path(**''**, views.travelo\_home, name=**'travel\_home'**),  
]

Now do mapping of this urls.py file of travel to main project urls.py file.

Project/urls.py

**from** django.contrib **import** admin  
**from** django.urls **import** path, include  
  
urlpatterns = [

# This mapping specific to travel  
 path(**'travel/'**,include(**'travelo.urls'**)),  
 path(**''**,include(**'calc.urls'**)),  
 path(**'contact/'**,include(**'mango.urls'**)),  
 path(**'admin/'**, admin.site.urls),  
  
]

* If All setting done then restart the server and go to url [**https://127.0.0.1:8000/travelo**](https://127.0.0.1:8000/travelo) **then travelo**  index page will be visible , but not in proper format
* To make it in proper format create a **static** folder in the project and copy all static files from downloaded template to inside it **static** folder in project. Like images, javascript etc.
* Now do static folder mapping in project **setting.py** file to make to accessible from **index.html**.

**Project/setting.py**

*# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/2.2/howto/static-files/*STATIC\_URL = **'/static/'**STATICFILES\_DIRS = [  
 os.path.join(BASE\_DIR, **'static'**)  
]  
STATIC\_ROOT = os.path.join(BASE\_DIR, **'assets'**)

# This will create assets folder inside the project and contains all data required for static mapping.

* To create folder inside the project we have pass a command to create a folder.

**python manage.py collectstatic**

* Once apply this command verify assets folder should be created.
* Now edit the index.html file and do some changes in all the urls and load the using command.

On top of html file : {% load static %}

And for all the path of css file in link :

Previous:

<**link rel="stylesheet" type="text/css" href=" plugins/OwlCarousel2-2.2.1/owl.carousel.css'"**>

After Edit:

<**link rel="stylesheet" type="text/css" href="{% static 'plugins/OwlCarousel2-2.2.1/owl.carousel.css' %}"**>

* Refresh the URL , Now template should properly on web browser.

Cre

**:Dynamic Content to the Website:**

1. Create a model class inside the **models.view**, this class take three parameters.

Model.py

**class** Destination:  
 name : str  
 img : str  
 prince : int  
 desc : str

1. Create view.py file and create destination class object.

dest1 = Destination()  
  
dest1.name = **'Mumbai'**dest1.price = **'800'**dest1.desc = **"The city which never sleep"**

dest2.name = **'Jaipur'**dest2.price = **'900'**dest2.desc = **"The Pink City"**dest3.name = **'Delhi'**dest3.price = **'100'**dest3.desc = **"Heart Of India"**

**def** index(request):  
 **return** render(request, **'index.html'**, {**'dest1'** : dest1, **'dest2'** : dest2, **'dest3'** : dest3})

1. Make changes in templates/index.html file as ninja template.

**Index.html**

<**html lang="en"**>  
<**head**>  
<**title**>Travello</**title**>  
{% load static %}

**.**

**.**

**.**

*<!-- Destinations -->*<**div class="destinations" id="destinations"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col text-center"**>  
 <**div class="section\_subtitle"**>simply amazing places</**div**>  
 <**div class="section\_title"**><**h2**>Popular Destinations</**h2**></**div**>  
 </**div**>  
 </**div**>  
 <**div class="row destinations\_row"**>  
 <**div class="col"**>  
 <**div class="destinations\_container item\_grid"**>  
  
 *<!-- Destination -->* <**div class="destination item"**>  
 <**div class="destination\_image"**>  
 <**img src="{% static 'images/destination\_1.jpg' %}" alt=""**>  
 <**div class="spec\_offer text-center"**><**a href="#"**>Special Offer</**a**></**div**>  
 </**div**>  
 <**div class="destination\_content"**>  
 <**div class="destination\_title"**><**a href="destinations.html"**>{{ dest1.name }}</**a**></**div**>  
 <**div class="destination\_subtitle"**><**p**>{{dest1.desc }}</**p**></**div**>  
 <**div class="destination\_price"**>From ${{ dest1.price }}</**div**>  
 </**div**>  
 </**div**>  
  
 *<!-- Destination -->* <**div class="destination item"**>  
 <**div class="destination\_image"**>  
 <**img src="{% static 'images/destination\_2.jpg' %}" alt=""**>  
 </**div**>  
 <**div class="destination\_content"**>  
 <**div class="destination\_title"**><**a href="destinations.html"**>{{dest2.name}}</**a**></**div**>  
 <**div class="destination\_subtitle"**><**p**>{{ dest2.desc }}</**p**></**div**>  
 <**div class="destination\_price"**>From ${{ dest2.price }}</**div**>  
 </**div**>  
 </**div**>  
  
 *<!-- Destination -->* <**div class="destination item"**>  
 <**div class="destination\_image"**>  
 <**img src="{% static 'images/destination\_3.jpg' %}" alt=""**>  
 </**div**>  
 <**div class="destination\_content"**>  
 <**div class="destination\_title"**><**a href="destinations.html"**>{{ dest3.name }}</**a**></**div**>  
 <**div class="destination\_subtitle"**><**p**>{{ dest3.desc }}</**p**></**div**>  
 <**div class="destination\_price"**>From ${{ dest3.price }}</**div**>  
 </**div**>  
 </**div**>

1. Refresh the browser all the dynamic changes should be visible on browser.

**FOR LOOPING AND IF CONDITION IN TEMPLATE**

**home.html**

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
<**title**>Travello</**title**>  
{% load static %}  
{% static "images" as baseUrl %}

.

.

.

.

{% for dest in dests %}  
  
 <**div class="destination item"**>  
 <**div class="destination\_image"**>  
 <**img src="{{ baseUrl }}/{{ dest.img }}" alt=""**>  
 {% if dest.offer %}  
 <**div class="spec\_offer text-center"**><**a href="#"**>Special Offer</**a**></**div**>  
 {% endif %}  
 </**div**>  
  
 <**div class="destination\_content"**>  
 <**div class="destination\_title"**><**a href="destinations.html"**>{{ dest.name }}</**a**></**div**>  
 <**div class="destination\_subtitle"**><**p**>{{ dest.desc }}</**p**></**div**>  
 <**div class="destination\_price"**>From ${{ dest.price }}</**div**>  
 </**div**>  
</**div**>  
  
{% endfor %}

**View.py**

**from** django.shortcuts **import** render  
**from** .models **import** Destination  
*# Create your views here.*dest1 = Destination()  
dest2= Destination()  
dest3 = Destination()  
  
dest1.name = **'Mombai'**dest1.price = **'800'**dest1.desc = **"The city which never sleep"**dest1.img = **'destination\_1.jpg'**dest1.offer = **False**dest2.name = **'Jaipur'**dest2.price = **'900'**dest2.desc = **"The Pink City"**dest2.img = **'destination\_2.jpg'**dest2.offer = **True**dest3.name = **'Delhi'**dest3.price = **'100'**dest3.desc = **"Heart Of India"**dest3.img = **'destination\_3.jpg'**dest3.offer = **False***# def home(request):  
# return render(request, 'index.html')*  
dests = [dest1, dest2, dest3]

**def** home(request):  
 **return** render(request, **'home.html'**, {**'dests'**: dests})

**urls.py**

**from** django.contrib **import** admin  
**from** django.urls **import** path, include  
**from** . **import** views  
  
urlpatterns = [  
 path(**''**, views.home, name=**'home'**),  
 path(**'index/'**, views.index, name=**'index'**),  
]

**DATABASE CONNECTION**

**Download Postgrey :** <https://www.enterprisedb.com/software-downloads-postgres>

**Download pgAdmin** : <https://www.postgresql.org/ftp/pgadmin/pgadmin4/v4.9/windows/>

Username : postgrey

Password : 1234

Port : 5432

Access Postgres : 127.0.0.1:56688/browser