In Django, a template is an HTML file (or other text-based format) that uses special syntax to dynamically insert content and logic. Templates allow you to separate the presentation layer (what the user sees) from the business logic layer (how data is processed and handled).

Django templates use the Django Template Language (DTL) syntax, which includes variables, filters, tags, and template inheritance. This syntax allows you to display dynamic content, loop through data, use conditional statements, and reuse layout elements across pages.

Basic Django Template Syntax:

Variables: Used to display dynamic data. Variables are surrounded by double curly braces {{ }}.

```
<p>Hello, {{ user.name }}!</p>
```

1.

Filters: Applied to variables to transform the data. Filters are added after the pipe | symbol within the curly braces.

```
<p>{{ user.name|upper }}</p>  <!-- Converts user.name to uppercase -->
```

2. Tags: Used for logic, such as loops and conditional statements, and are surrounded by {% %}.

For loop: Loops over a list of items.

```
{% for item in items %}
  <p>{{ item.name }}</p>
{% endfor %}
```

If statement: Conditionally displays content.

```
{% if user.is_authenticated %}
  <p>Welcome back, {{ user.name }}!</p>
{% else %}
  <p>Hello, guest!</p>

{% endif %}
```

Key Points:

- Variables ({{ }}) display dynamic data.
- Filters (|) transform data within variables.
- Tags ({% %}) provide logic for loops, conditionals, and template structure.
- Variables are written in double curly braces {{ }}, e.g., {{ user.name }}.
- Filters are used with a pipe symbol | to format variables, e.g., {{ user.name|upper }}.
- Tags provide template logic within {% %}, like {% for %} and {% if %}.

What Are Static Files?

Static files are files like images, CSS, and JavaScript that are used to style and add interactivity to web pages. They are called "static" because they are served directly to the browser without server-side processing. Static files are essential for creating a visually appealing and user-friendly web experience.

How Static Files Work in Django?

Django has a system to help you manage and serve these static files.

1. STATIC_URL: This setting tells Django the URL where static files will be accessible to users. In most cases, it is set to /static/.
    - Example: If you have a CSS file style.css, it will be accessible to the user at http://yourwebsite.com/static/css/style.css.

    python
    Copy code
    STATIC_URL = '/static/'

2. STATICFILES_DIRS: This setting tells Django where to find static files during development. You can create a static/ folder in your project and add your CSS, JavaScript, and images there.
    - Example: If your project is in the folder myproject/, you can create a static/ folder in the same directory.

python
Copy code

```
STATICFILES_DIRS = [
    BASE_DIR / "static",
]
```

3. STATIC_ROOT: This is where Django collects all static files when you prepare your project for production (live website). This helps organize static files into one place for easy delivery.
    - Example: If your project is live, you run the collectstatic command to gather static files in one location so your web server can serve them quickly.

python
Copy code

```
STATIC_ROOT = BASE_DIR / "staticfiles"
```

To use static files in a template:

1. First, load the static template tag at the top: {% load static %}.
2. Use {% static 'path/to/file' %} to reference the file in the HTML. For example:

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

<img src="{% static 'images/logo.png' %}" alt="Logo">
```

Project:

Project creation: django-admin startproject tempandstatic

App creation: django-admin startapp myapp

In app: templates folder creation

In app: static folder creation

Project - settings.py file:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp'
]
STATIC_URL = 'static/'
STATICFILES_DIRS=[
    os.path.join(BASE_DIR,'static')
]
```

App - views.py file:

```
from django.shortcuts import render
import datetime


# Create your views here.
def wish(request):
    time=datetime.datetime.now()
```

```
    name='ABC'
    rollno=101
    marks=90
    format_time=time.strftime('%d-%m-%y %H:%M:%S')
    my_dict={'insert_date':format_time,'name':name,'rollno':rollno,'marks':marks}
    return render(request,'wish.html',my_dict)


def wish1(request):
    time=datetime.datetime.now()


    msg="Hello Everyone"
    h=int(time.strftime('%H'))
    if h < 12:
        msg +='Good Morning'
    elif h < 16:
        msg +='Good Afternoon'
    elif h < 21:
        msg +='Good evening'
    else:
        'Good night'
    my_dict={'insert_time':time,'insert_data':msg}
```

```
    return render(request,'wish1.html',context=my_dict)


def greet(request):
    time = datetime.datetime.now()
    current_time = time.strftime('%H:%M:%S')
    if time.hour < 12:
        greeting = "GoodMorning"
    elif time.hour < 16:
        greeting = "GoodAfternoon"
    else:
        greeting = "GoodEvening"
    return render(request, 'greet.html', {'greeting': greeting, 'current_time':
current_time})
```

Create a template in application:

1. wish.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
```

```
</head>
<body>
  <h1>Wish page</h1>
  <h2>Data from first page</h2>
  <p>Insert time:{{insert_date}}</p>
  <p>Name:{{name}}</p>
  <p>Roll no:{{rollno}}</p>
  <p>Marks:{{marks}}</p>

</body>
</html>
```

2. wish1.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Wish1 page</h1>
```

```
    <h2>Data from Second page</h2>
    <p>Insert time:{{insert_time}}</p>
    <p>Insert data:{{insert_data}}</p>



</body>
</html>
```

3. greet.html:

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
    <script src="{ static 'js/script.js'}"></script>
</head>
<style>
    body{
        background-color: aqua;
```

```
        padding-left: 400px;
    }
</style>
<body>
    <h2>Greet Page</h2>
    <div class="container">
        <img src="{% static 'images/nature.jpeg' %}"alt="Nature">
        <h2>Time:{{current_time}}</h2>
        <h2>Greeting:{{greeting}}</h2>
    </div>
    <script>
        document.write('Hello everyone,i am from js')
    </script>


</body>
</html>
```

Create a static folder in app, in that we have created folders css, images, js:

For css: `<link rel="stylesheet" href="{% static 'css/styles.css' %}">`

For js: `<script src="{ static 'js/script.js'}"></script>`

For images: `<img src="{% static 'images/nature.jpeg' %}"alt="Nature">`

Application level urls.py file:

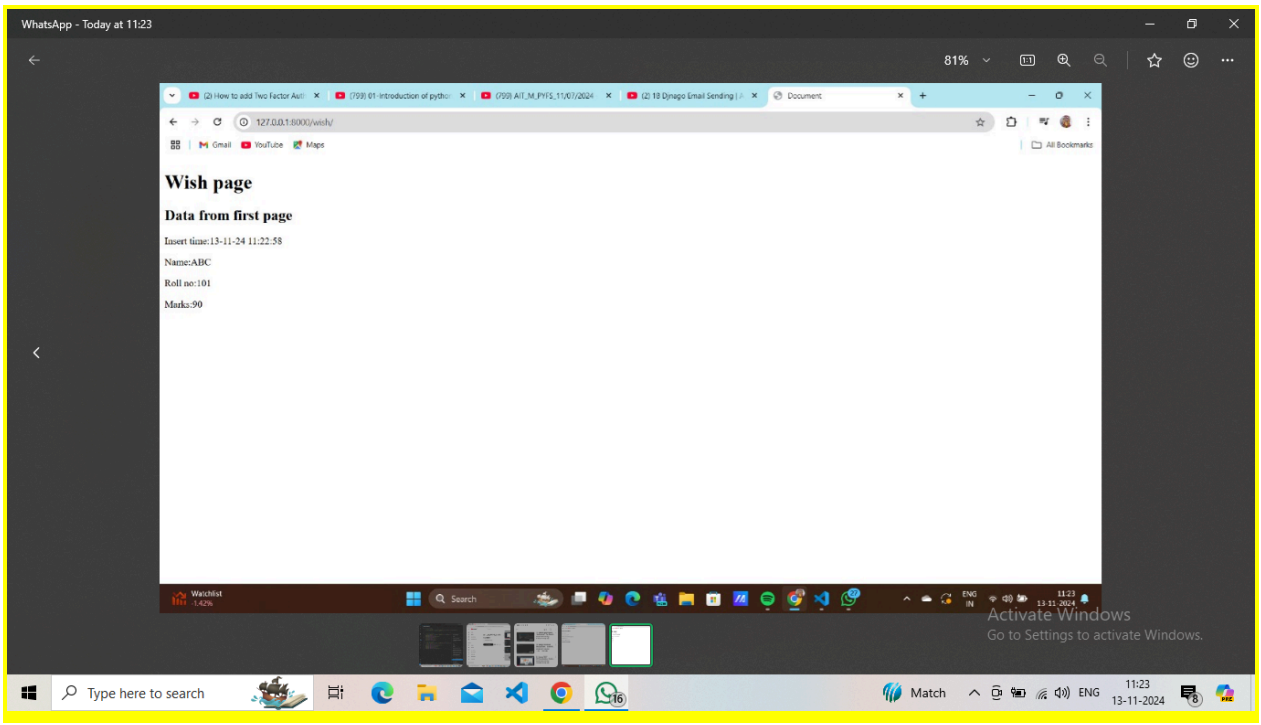
```
from django.urls import path
from .import views
urlpatterns=[
    path('wish/',views.wish),
    path('wish1/',views.wish1),
    path('greet/',views.greet),
]
```

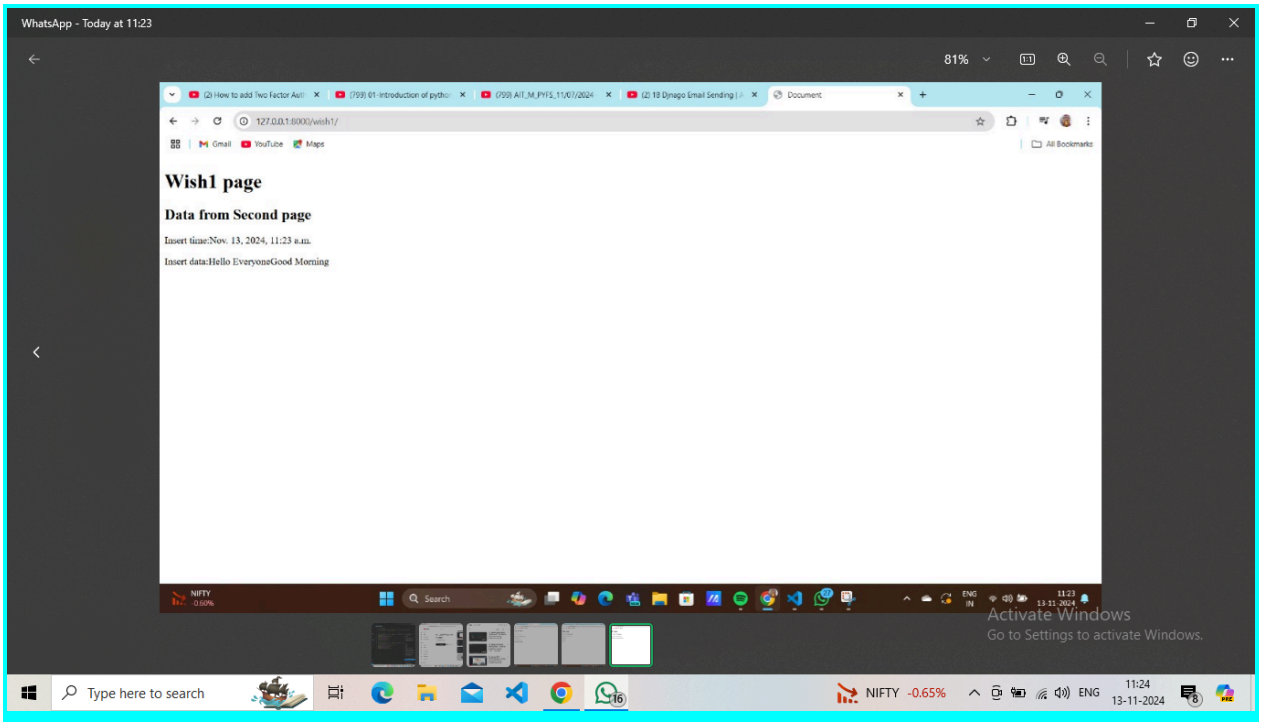
Project level urls.py file:

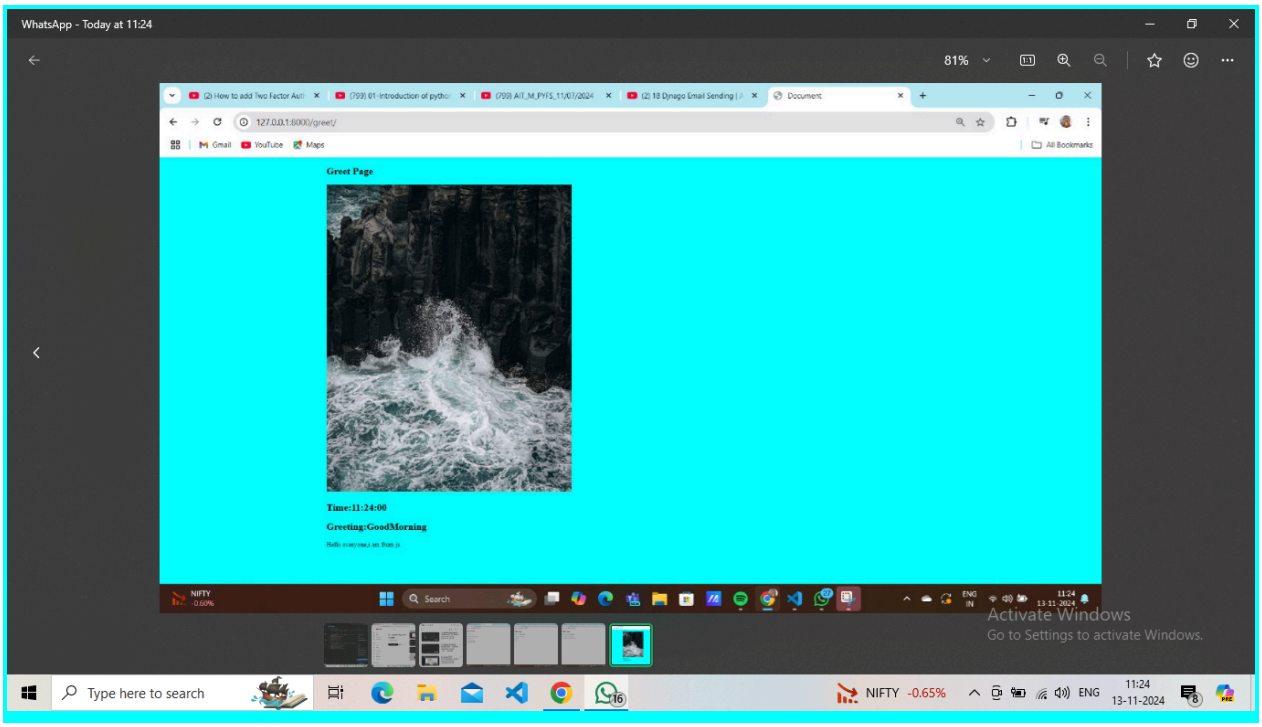
```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('myapp.urls')),
]
```

```
Python manage.py runserver
```

Output:


WhatsApp - Today at 11:23
127.0.0.1:8000/wish/
Wish page
Data from first page
Insert time:13-11-24 11:22:58
Name:ABC
Roll no:101
Marks:90
Activate Windows
Go to Settings to activate Windows.
Type here to search
Match
ENG
11:23
13-11-2024


WhatsApp - Today at 11:23
127.0.0.1:8000/wish1/
Wish1 page
Data from Second page
Insert time:Nov. 13, 2024, 11:23 a.m.
Insert data:Hello EveryoneGood Morning
Activate Windows
Go to Settings to activate Windows.
Type here to search
NIFTY -0.65%
ENG
11:24
13-11-2024


WhatsApp - Today at 11:24
81%
127.0.0.1:8000/greet/
Gmail
YouTube
Maps
Greet Page
Time:11:24:00
Greeting:GoodMorning
Activate Windows
Go to Settings to activate Windows.
Type here to search
NIFTY -0.65%
ENG
11:24
13-11-2024