

Django Project Documentation:

This documentation explains how the Django project is structured and how each component works. The project consists of Django application named 'myapp', which has three views: 'index', 'peoples-info' and 'peoples info details'.

Below is the breakdown of how each part of the project function.

* Project structure:

~~myse~~

mysite/

– mysite/

 | – settings.py

 | – urls.py

– myapp/

 | – views.py

 | – urls.py

 | – templates/

 | index.html

 | peoples-info.html

 | peoples-info-details.html

① Project configuration: ('mysite/settings.py')

The 'Installed-apps' section in 'settings.py' registers all the built-in Django apps and the custom app 'myapp':

"Python

INSTALLED_APPS = [

 'django.contrib.admin',

 'django.contrib.auth',

 'django.contrib.contenttypes',


```
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',  
'myapp',  
]
```

'''

this ensures that django recognizes 'myapp' and allow it to function properly.

② URL Configuration:

Django uses 'urls.py' files to define how URLs are mapped to views.

Project level URL Configuration (mysite/urls.py)

```
''' python
```

```
from django.contrib import admin
```

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

```
url_patterns = [
```

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

```
    path('', include('myapp.urls')),
```

```
]
```

```
,
```

- The 'path('admin/', admin.site.urls)' allows access to django's admin panel.

App-level URL configuration ('myapp/urls.py')

```
''' python
    from django.urls import path
    from .views import *
```

```
urlpatterns = [
    path("", index),
    path('peoples-info/', peoples_info),
    path('peoples-info-detail/', peoples_info_detail),
]
```

- '/' → 'index' views
- '/ peoples-info/' → 'peoples-info' view
- '/ peoples-info-detail/' → 'peoples-info-detail' view

3' Views ('myapp/views.py')

Django view handle request and return responses.
In this project, three views are defined.

3.1 'index' view

```
''' python
```

```
from django.shortcuts import render
name = 'q2q2'
age = 30
job = 'developer'
```



```
def index(request):
    print(name)
    return render(request, 'index.html', {'name': name,
    'age': age, 'job': job})
'''
```

- Defines three variables : 'name', 'age' and 'job'.
- prints the value of 'name' ('azad') to the console when the view is accessed
- passes 'name', 'age' and 'job' as context data to 'index.html',

3.2 'peoples-info' View

```
''' python
peoples = [
    {'name': 'azad', 'age': 30, 'job': 'developer'},
    {'name': 'sayjad', 'age': 25, 'job': 'designer'},
    {'name': 'mamun', 'age': 35, 'job': 'manager'},
    {'name': 'tarek', 'age': 40, 'job': 'teacher'},
]
```

```
def peoples_info(request):
```

```
    print('ky chal mai aya')
```

```
    return render(request, 'peoples-info.html',
    {'peoples': peoples})
'''
```


- Defines a list 'peoples' containing dictionaries with details of four individuals.
- prints 'tu chal mai aayd' to the console when accessed.
- Passes the 'peoples' list as context data to 'peoples-info.html';

④ Template :

Django templates are used to render dynamic HTML content.

4.1 'index.html'

```
''' = 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>
```

```
this is a2ad from nowhere
```

```
{{ name | upper }} <!-- converts name to uppercase -->
```

```
{{ nam }} <!-- error
```

```
</body>
```

```
</html>
```

```
'''
```


- Displays "this is a2ad from nowhere" ,
- Uses '{{ name | upper }}' to display 'AZAD'
- '{{ nam }}' will cause an error because 'nam' is not defined in the context.

4.2 ' peoples-info.html '

```

''' 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>
  <ul>
    {% for p in peoples %}
    <li> name: {{ p.name }} </li>
    <li> age: {{ p.age }} </li>
    <li> Job: {{ p.job }} </li>
    <hr>
    {% endfor %}
  </ul>

</body>
</html>
'''

```


- Loops through the 'peoples' list and displays each person's name, age, and job in an unordered list.

4.3 'peoples - Info - detail. html'

```
''' 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>
```

```
{ % for k, v in details.items % }
```

```
<h1> {{ k }} </h1>
```

```
<p> {{ v }} </p>
```

```
{ % endfor % }
```

```
</body>
```

```
</html>
```

```
'''
```

- Iterate through the 'details' dictionary.
- Displays each key ('k') as a heading and the corresponding dictionary ('v') as a paragraph.
- However, '{{ v }}' will not be properly formatted because it is a dictionary.

It should be accessed as " {{ v.name }}",
{{ v.age }}', and {{ v.job }}'.

⑤ Issues in the code

1. Undefined Variables in 'index.html'

```

<< html
  {{ nam }}
  </html>

```

- nam is not defined in the view context, which will cause an error.

~~incorrect~~

Incorrect Display of Dictionary in 'Peoples-info-detail.html'

```

<< html
  <p>{{ v }} </p>
  </html>

```

- Should be replaced with :

```

<< html
  <p> name : {{ v.name }} </p>
  <p> Age : {{ v.age }} </p>
  <p> Job : {{ v.job }} </p>
  </html>

```