

## **1) Explain the concept of including URLs using the include function.**

Include() function in django used to include the URL pattern from another file in the project. We can add multiple URL patterns using include() function. By using include() function, we can reduce the code repetition and we can minimize the code.

To use the include function pass the path to the file that contains the url patterns.

## **2) What is the significance of the render function in Django views?**

render() function is used to render a template into a string that is returned as a HttpResponse. To render a template steps included are follows.

- Import render() function from django.shortcuts
- In view function pass the request object, template name, and context dictionary to the render(), context dictionary contains the data that you want to pass to the template for rendering.
- render() function process the template generates http response.

render() function performs a very important part in django as it allows generating web pages by combining python data and HTML templates.

## **3) What are some common field types in models ?**

CharField - Used for small to large sized strings. we can specify maximum length.

TextField - Used for large amounts of text. It doesn't have maximum length

IntegerField - Used for storing whole numbers.

FloatField - Used for storing floating point numbers

BooleanField - Used for storing Boolean numbers.

DateField - Used for storing date.

DateTimeField - Used for storing date and time.

FileField - Used for storing files.

ImageField - Used for storing images.

EmailField - Used for storing Email addresses.

ForeignKey - A field that reference another model

OneToOneField - A field references exactly one model.

ManyToMany - A field references multiple models.

#### **4) Explain the purpose of the name parameter in the path function when defining URLs**

Name parameter in path function used to give name to the URL pattern. Naming URL pattern is helpful when referring to the url in the template. Name parameter in the path() helps to understand the code in a more human readable way. We can call or use the url patterns by referring the name parameter and its is easy to change the URLs in the future without updating the links throughout the project.

#### **5) Describe the difference between class-based views and function-based views in Django. Give examples.**

They are the two approaches for defining views in django.

##### Class -based views

-Class based views inherited from django's generic view class.

-They are easy to reuse and extend using sub classing.

-It provides a consistent interface for handling HTTP requests.

-It's more modular and it can be easily broken down into small and manageable components.

Example:

```
from django.views.generic import TemplateView
```

```
class IndexView(TemplateView):
```

```
    template_name = 'index.html'
```

### Function Based Views

-It's implemented as simple python functions.

-As it handle specific url pattern its less reusable

-The code functionality depends upon the view created as the view created becomes complex and the code also becomes longer.

-It's defined in views.py file.

Example:

```
from django.http import HttpResponse
```

```
def mya_pp(request):
```

```
    return HttpResponse('Hello, world!')
```

## **6) What role do template tags play in Django templates, and provide examples of commonly used template tags**

Template tags are used in django to extend functionality of django templates.

Template tags are written in python, are enclosed in `{% %}` for control statements and `{{ }}` for variable rendering.

Template tags also help to change the logic in views without affecting the structure of templates.

`{% for %}` and `{% endfor %}` - Used to iterate over a list or other iterable object.

`{% if %}` and `{% endif %}` - Used to execute conditional logic.

`{% else %}` and `{% elif %}` - Used to provide alternative paths for conditional logic.

`{% extends %}` and `{% include %}` - Used to extend and include other templates.

`{% url %}` - Used to generate URLs.

`{% static %}` - Used to generate static asset URLs.