# 1. Explain Django Architecture?

Ans:

**Django**, a Python framework to create web applications, is based on Model-View-Template (MVT) architecture. **MVT** is a software design pattern for developing a web application.

* Model: A **Model** is an object that defines the structure of the data in the Django application.
* View: A **View** is a handler function that accepts HTTP requests, processes them, and returns the HTTP response.
* Template: A **Template** is a text file that defines the structure or layout of the user interface.

# 2. Explain the Django project directory structure?

Ans:

* manage.py- This file is used to interact with your project via the command line(start theserver, sync the database…etc.).

eg: Python manage.py runserver

* \_init\_.py – It is a python package. It is invoked when the package or a module in the package is imported.
* settings.py – As the name indicates it contains all the website settings. In this file, we register any applications we create, the location of our static files, database configuration details, etc.
* urls.py – In this file, we store all links of the project and functions to call.
* wsgi.py – This file is used in deploying the project in WSGI. It is used to help your Django application communicate with the webserver.

# 3. What are models in Django?

Ans:

A model is a class that represents table or collection in our DB, and where every attribute of the class is a field of the table or collection. Models are defined in the app/models.py (in our example: myapp/models.py)

# 4. What are templates in Django or Django template language?

Ans:

In the Django MTV model, the template is the presentation layer. This layer interacts with the user, sends the requests to the views, and responds to the user.

Django Template Language or DTL is **a text-based Template language that provides a bridge between scripts like HTML, CSS, JS, etc.** **and programming languages like python**. DTL is specifically built for developers to embed Django logic codes into HTML template files.

# 5. What are views in Django?

Ans:

A **Template** is a text file that defines the structure or layout of the user interface. The text file can be any type of file. for example: HTML, XML, etc.

It can accept data from the view and render it using jinja syntax.

# 6. What is Django ORM?

Ans:

Django ORM is one of the best tools of Django and plays very essential role to perform database related tasks. It provides abstractions with the database, in a mostly database agnostic way.

# 7. Define static files and explain their uses?

Ans:

Websites generally need to serve additional files such as images, JavaScript, or CSS. In Django, we refer to these files as “static files”. Django provides [**django.contrib.staticfiles**](https://docs.djangoproject.com/en/4.1/ref/contrib/staticfiles/#module-django.contrib.staticfiles) to help you manage them.

# 8. What is Django Rest Framework (DRF)?

Ans:

Django REST framework (DRF) is **a powerful and flexible toolkit for building Web APIs**. Its main benefit is that it makes serialization much easier. Django REST framework is based on Django's class-based views, so it's an excellent option if you're familiar with Django.

# 9. What is Django-admin and manage.py and explain its commands?

Ans:

Django-admin is Django’s command-line utility for administrative tasks. This document outlines all it can do.  
   Eg:- Django-admin startproject demoproject    
  
   Manage.py is used to interact with project via the commands line(startserver,sync database…)  
   eg: python manage.py runserver

# 10. What is Jinja templating?

Ans:

Jinja templates offer some basic programming functionalities such as variable substitutions, for loops, functions calls, filters, as well as the ability to extend base components.

jinja templating syntax provide set of rules  and constructs that define how templating language should be used and how the output should be generated.

# 11. What are Django URLs?

Ans:

Store all links of the project and functions to call.Every page on the Internet needs its own URL.This way your application   knows what it should show to a user who opens that URL.  
 - In Django, we use something called URLconf (URL configuration).  
 - URLconf is a set of patterns that Django will try to match the requested URL to find the correct view.

# 12. What is the difference between a project and an app in Django?

Ans:

## **project:-**

* A project in Django is a python package that represents the whole web application.
* A project in Django basically contains the configuration and setting related to the entire website.
* A single project can also have multiple apps in it that can be used to implement some functionality.

**App:-**

* An app in Django is a sub-module of a project, and it is used to implement some functionality.
* Now, you can refer to an app as a standalone python module that is used to provide some

functionality to your project.

* We can create multiple apps within a single Django project. And these apps can be independent of one another. Theoretically, we can use an app from one Django project to another without making any changes to it.

# 13. What are different model inheritance styles in the Django?

Ans:

**1. Abstract Base Class Inheritance:**

**This style is used when you only want parents class to hold information that you don’t want to type out for each child model.**

# 2.Multi Table Model Inheritance:

**This style is used if you are sub-classing an existing model and need each model to have its own database table.**

# 3. Proxy Model Inheritance:

**This style is used, if you only want to modify the Python level behaviour of the model, without changing the model’s fields.**