DJANGO

1. What is Django?

Django is a Full-stack web development framework that facilitates the creation and maintenance of high-quality Dynamic pages while also encouraging rapid development and a clean, pragmatic style. Django makes it easier to automate repeated operations, resulting in a more efficient development process with fewer lines of code.

2.Name some Companies that use Django.

Some companies that use the Django framework are Instagram, Mozilla, Disqus, Bitbucket, Nextdoor, and Clubhouse.

3.What is the difference between a Project and an App?

The main difference between a project and an app is that a project is defined as the entire application whereas, an app is part of the project that is self-sufficient to perform any task.

4. Explain Django's architecture.

The Model-View-Template also known as MVT architecture is used by Django. It is a software design pattern for developing a web application. The Django MVT Structure is made up of three parts:

- The model will serve as an interface for your data. It is in charge of data management. A database represents the logical data structure that supports the entire application such as MySql and Postgres.
- The View is the user interface, that renders a website page in your browser. HTML/CSS/Javascript and Jinja files are used to represent it.
- A template is made up of both static sections of the desired HTML output and specific syntax that describes how dynamic content will be included.

6. What are the Features of using Django?

- Flexible server arrangement,
- Model relation database
- Provide object-relational mapper
- Web templating system

- Middleware class support

- Regex-based URL Dispatcher

- Unit testing framework

- Admin Interface

- Django is SEO optimized.

- In-build mitigation

- Easy inheritance

7.Explain the Django project directory structure.

When you first start a Django project, it comes with some basic files like manage.py and view.py.

1. init.py – It's an empty Python file. It is called when the package or one of its modules is imported. This file tells the Python interpreter that this directory is a package and that the presence of the __init.py_ file makes it a Python project.
2. manage.py – This file is used to interact with your project from the command line utility. with the help of this command, we can manage several commands such as:
   1. manage.py runserver
   2. manage.py makemigration
   3. manage.py migrate' etc
3. setting.py – It is the most important file in Django projects. It holds all the configuration values that your web app needs to

work, i.e. pre-installed, apps, middleware, default database, API keys, and a bunch of other stuff.

4. views.py – The View shows the user the model's data. The view knows how to get to the data in the model, but it has no idea what that data represents or what the user may do with it.

5. urls.py – It is a universal resource locator which contains all the endpoints, we store all links of the project and functions to call it.

6. models.py – The Model represents the models of web applications in the form of classes, it contains no logic that describes how to present the data to a user.

7. wsgi.py – WSGI stands for Web Server Gateway Interface, This file is used for deploying the project in WSGI. It helps communication between your Django application and the web server. more…

8. admin.py – It is used to create a superuser Registering model, login, and use the web application.

9. app.py – It is a file that helps the user to include the application configuration for their app.

8. How do you create a Django project?

We can create a Django project with the help of the following command

django-admin startproject projectname

9. How do you create a Django app?

We can create a Django app with the help of the following command

python manage.py startapp appname

10. How do we start our development server?

We can start our development server with the help of the following command

python manage.py runserver

11. Importance of virtual environment setup for Django.

A virtual environment allows you to establish separate dependencies of the different projects by creating an isolated environment that isn't related to each other and can be quickly enabled and deactivated when you're done.

It is not necessary to use a virtual environment without a virtual environment we can work with Django projects. However, using virtualenv is thought to be the ideal practice. Because it eliminates dependencies and conflicts.

12.Give a brief about the Django admin interface.

Django provides us with a default admin interface that is very helpful for creating, reading, updating, and deleting model objects that allow the user to do administrative tasks. It reads a set of data that explains and provides information about data from the model

in order to create a quick interface where the user can alter the application's contents. This is an in-built module.

13. What are Django URLs?

The routing of a website is determined by its URLs. In the program, we build a python module or a file called urls.py. The navigation of your website is determined by this file. When a user visits a given URL route in the browser, the URLs in the urls.py file are compared. The user then receives the response for the requested URL after retrieving a corresponding view method.

14. What are the views of Django?

Django views are an important feature of the MVT Structure. This is a function or class that takes a web request and delivers a Web response, according to the Django script. This response could be an HTML template, a content of a Web page, an XML document, a PDF, or images. the view is a part of the user interface that renders the HTML/CSS/Javascript in your Template files into what you see in your browser when we render a web page.

15. What are the models in Django?

Model is a built-in feature of Django that contain a definitive source of information such as behavior and essential fields about the data we are storing. It allows us to build tables, fields, and constraints and organize tables into models. Generally, each model maps to a single database table. To use Django Models, you'll need a project and an app to work with. Also, Django makes use of SQL to access the database. SQL is a complex language with many

queries for generating, removing, updating, and other database-related tasks.

16. What do the following commands do?
- python manage.py makemigrations
- python manage.py migrate

The makemigration command scans the model in your application and generates a new set of migrations based on the model file modifications. This command generates the SQL statement, and when we run it, we obtain a new migration file. After running this command, no tables will be created in the database.
Running migrate command helps us to make these modifications to our database. The migrate command runs the SQL instructions (produced by makemigrations) and applies the database changes. After running this command, tables will be created.

17. Define static files and explain their uses.

Static files are used to save files such as CSS, JavaScript, pictures, and other types of static files. We keep them in distinct folders, for as the js folder, which has all of the JavaScript files, and the images folder, which contains all of the images. These files are kept in the static subfolder of the project app. Django provides django.contrib.staticfiles which helps us to manage static files. There are different uses for static files:
- It clarifies the use of file methods and attributes.
- It is platform-independent in many ways, whereas generic files are not.
- Subclasses can be used to extend it.

18.What are templates in the Django language?

A Django template is a text document that is used to give a front and a layout for our website. It is the third and most significant aspect of Django's MVT Structure. In Django, a template is an HTML file that contains HTML, CSS, and Javascript. The Django framework efficiently manages and generates dynamically generated HTML web pages for end-user viewing. Django is mostly a backend framework, thus we use templates to give a layout for our website. There are two ways to incorporate the template into our website.

- We can utilize a single template directory that will be distributed throughout the project.
- We can make a separate template directory for each app in our project.

19.What does the settings.py file do?

settings.py is a core file in Django projects. It holds all the configuration values that your web app needs to work; database settings, logging configuration, where to find static files, API keys if you work with external APIs, and a bunch of other stuff.

21. Difference between MVC and MVT design patterns?

- Model and View are both driven by the controller in MVC, whereas Views in MVT are used to receive HTTP requests and return HTTP responses.
- We must write all of the control-specific code in MVC whereas, We must write all of the control-specific code in the View and Template components in MVT.
- MVC is Highly coupled whereas, MVT is loosely coupled.
- In MVC, it is difficult to modify whereas, Modification is easy in MVT.
- MVC is suitable for large applications, but MVT is suitable for both small and large applications.
- MVC does not involve any URL mapping, whereas in MVT URL pattern mapping takes place.
- Flow is clear and easy to understand, whereas MVT is sometimes harder to understand.