

1) What is Django?

→ Django is a free and open-source web application framework, written in Python.

→ A web framework is a set of components that helps you to develop websites faster and easier.

2) Difference between Django and flask?

→ Both Django and Flask are popular frameworks of Python. Each framework has a unique quality, and we can use it according to the project requirement.

→ Django is a full-stack web framework, which is used for large and complex web applications, whereas Flask is a lightweight and extensible web framework.

3) Which architectural pattern does Django follow?

→ 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.

4) What is admin.py and what is the Django admin interface?

→ The admin.py file is used to display your models in the Django admin panel. You can also customize your admin panel.

→ One of the most powerful parts of Django is the automatic admin interface.

→ It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site.

→ The admin's recommended use is limited to an organization's internal management tool. It's not intended for building your entire front end around.

5) Difference between MVT and MVC?

→ In MVC, programmers need to write all the control-specific code whereas, in MVT, the framework itself handles the controller part.

6) What is a model in Django?

→ A model is the single, definitive source of information about your data. It contains the essential fields and behaviors of the data you are storing in the database. Generally, each model maps to a single database table.

7) What is the main use of manage.py

→ It is your tool for executing many Django-specific tasks.

→ like, starting a new app within a project, running the development server, running your tests, etc.

8) What are migrations in Django?

→ To create a model in the database in Django migrations must be done.

→ Migrations are Django's way of propagating changes you make to your models (adding a field, deleting a model, etc.) into your database schema.

→ So, we can say that after every change in models migrations need to be done every time.

9) Where do we write business logic in Django?

→ The answer is the almighty business layer where our business rules live. Sometimes divided into subsets like domain logic and application logic, it is the layer where we shall implement our business flows, actions, and logic.

10) What is the use of urls.py?

→ In urls.py, the most important thing is the "urlpatterns" tuple.

→ It's where you define the mapping between URLs and views.

→ A mapping is a tuple in URL patterns like – from Django.conf.urls import patterns, include, url from Django.

11) What is the use of the include function in urls.py?

→ Using include () functions we can import the path to myapp in urls.py files in Django.

→ include () A function that takes a full Python import path to another URLconf module that should be "included" in this place.

→ Example --

```
urlpatterns = [  
    Path ( ' ', include ("myapp.urls")  
    ]
```

12) Why is Django called a loosely coupled framework

→ Django is called a loosely coupled framework because of its MVT architecture, which is a variant of the MVC architecture.

→ It helps in separating the server code from the client-related code.

13) What is the Django rest framework?

→ Django rest framework is a powerful and flexible toolkit for building Web APIs. Some reasons you might want to use the rest framework.

14) Explain the importance of settings.py and what it contains?

→ settings.py is a core file in Django projects.

→ It holds all the configuration values that your web app needs to work.

→ Example –

1. To register a Django application,
2. Database Setting,
3. Logging configuration,
4. Where to find static files,
5. API keys if you work with external APIs,
6. and a bunch of other stuff.

15) What is ORM?

→ Object Relational Mapping (ORM) is a technique used in creating a "bridge" between object-oriented programs and, in most cases, relational databases.

→ Put another way, you can see the ORM as the layer that connects object-oriented programming (OOP) to relational databases.

16) What is the render () function in Django?

→ render () function takes two arguments, HTML code, and an HTML element.

→ The purpose of the function is to display the specified HTML code inside the specified HTML elements.

17) What is Jinja Templating?

→

18) Why is Django better than flask?

- Django is suitable for multiple-page applications.
- Flask is suitable for only single-page applications.
- -Django-admin is the in-built bootstrapping tool of Django that allows the creation of web applications without any external input.
- Flask does not come with an in-built bootstrapping tool.

19) Explain user authentication in Django?

- The Django authentication system handles both authentication and authorization. Briefly, authentication verifies a user is whom they claim to be, and authorization determines what an authenticated user is allowed to do.
- Here the term authentication is used to refer to both tasks.

20) Explain Middleware in Django.

- Middleware is a framework of hooks into Django's request/response processing.
- It's a light, low-level “plugin” system for globally altering Django's input or output.
- Each middleware component is responsible for doing some specific function.

21) What is static in Django?

- Aside from the HTML generated by the server, web applications generally need to serve additional files — such as images, JavaScript, or CSS — necessary to render the complete web page. In Django, we refer to these files as “static files”.

22) Explain the Django session.

- Sessions are the mechanism used by Django (and most of the Internet) for keeping track of the "state" between the site and a particular browser.
- Sessions allow you to store arbitrary data per browser and have this data available to the site whenever the browser connects.

23) What are templates?

→ Django Templates. Django provides a convenient way to generate dynamic HTML pages by using its template system. A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

24) Explain the Difference between project and apps

→ The difference between a project and an app is, a project is a collection of configuration files and apps whereas the app is a web application that is written to perform business logic.

25) How to view and filter items from the database?

→ On the Home tab, in the Sort & Filter group, click Advanced and then click Advanced Filter/Sort on the shortcut menu. Add the fields on which you want to filter to the grid. In the Criteria row of each field, specify a criterion.

26) What do you mean by the csrf_token?

→ A CSRF token is a secure random token (e.g., synchronizer token or challenge token) that is used to prevent CSRF attacks. The token needs to be unique per user session and should be of large random value to make it difficult to guess. A CSRF secure application assigns a unique CSRF token for every user session.