**Top Answers to Django Interview Questions**

**1. What is Django?**

Django is a server-side web application development framework written in Python.

**2. Which architectural pattern does Django follow?**

Django follows the Model View Template (MVT) architecture. MVT is a slight variation of the Model View Controller (MVC) architecture.

**3. What is the difference between a project and an app in Django?**

In Django, a project is the entire application and an app is a module inside the project that deals with one specific requirement. E.g., if the entire project is an ecommerce site, then inside the project we will have several apps, such as the retail site app, the buyer site app, the shipment site app, etc.

**4. What is Django Admin Interface?**

Django comes with a fully customizable in-built admin interface, which lets us see and make changes to all the data in the database of registered apps and models. To use a database table with the admin interface, we need to register the model in the admin.py file.

**5. Explain Django's Request/Response Cycle.**

In the Request/Response Cycle, first, a request is received by the Django server. Then, the server looks for a matching URL in the urlpatterns defined for the project. If no matching URL is found, then a response with 404 status code is returned. If a URL matches, then the corresponding code in the view file associated with the URL is executed to build and send a response.

**6. What is a model in Django?**

A model is a Python class in Django that is derived from the django.db.models.Model class. A model is used in Django to represent a table in a database. It is used to interact with and get results from the database tables of our application.

**7. What are migrations in Django?**

A migration in Django is a Python file that contains changes we make to our models so that they can be converted into a database schema in our DBMS. So, instead of manually making changes to our database schema by writing queries in our DBMS shell, we can just make changes to our models. Then, we can use Django to generate migrations from those model changes and run those migrations to make changes to our database schema.

**8. What are views in Django?**

A view in Django is a class and/or a function that receives a request and returns a response. A view is usually associated with urlpatterns, and the logic encapsulated in a view is run when a request to the URL associated with it is run. A view, among other things, gets data from the database using models, passes that data to the templates, and sends back the rendered template to the user as an HttpResponse.

**9. What is the use of the include function in the urls.py file in Django?**

As in Django there can be many apps, each app may have some URLs that it responds to. Rather than registering all URLs for all apps in a single urls.py file, each app maintains its own urls.py file, and in the project's urls.py file we use each individual urls.py file of each app by using the include function.

**10. 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. Django's models and views take care of the code that needs to be run on the server like getting records from database, etc., and the templates are mostly HTML and CSS that just need data from models passed in by the views to render them. Since these components are independent of each other, Django is called a loosely coupled framework.

**11. What is Django ORM?**

ORM stands for Object-relational Mapper. Instead of interacting with the database by writing raw SQL queries and converting the data returned from the query into a Python object, ORM allows us to interact with the database using objects of our model class. So, we just interact with our models and ORM converts these changes into SQL queries based on the database we are using, e.g., SQLite.

**12. How do templates work in Django?**

In Django, templates are used to dynamically generate web content. Django's templating engine handles templating that involves parsing, processing, and converting the template into an HttpResponse to return back to the client. These templates are by default written in Django Templating Language (DTL), which allows us to output the dynamic content in the templates based on the data passed in by the view.

**13. How do we register a model with Django admin?**

To register a model with Django's admin interface, we make changes to our apps admin.py file. We have to open the admin.py file in the app folder in which our models are. For example, if we have an app named 'polls' and we wish to register a model named 'Question', then we need to open 'polls/admin.py' and import the Question model and write: admin.site.register(Question). This will register our Question model with the admin site.

**14. How do we generate a super user in Django?**

In our project folder that contains Django's manage.py script, we have to open the command prompt and type the **python manage.py createsuperuser** command. Then, we need to enter the username, the email, and finally the password, twice (for conformation). This will create a super user for our Django project.

**15. Mention some disadvantages of Django.**

Django has the following disadvantages:
- Django' modules are bulky.
- It is completely based on Django ORM.
- Components are deployed together.
- We must know the full system to work with it.

**16. What is Sessions Framework in Django?**

The Sessions framework in Django is used to store arbitrary information about the user on the server in the database. This is done because HTTP is a stateless protocol, i.e., it does not store information between subsequent requests. Django uses a cookie containing a special session ID to identify each browser and its associated session with the site.

**17. What is a cookie in Django?**

A cookie is a small piece of information that is stored in the client browser. It is used to store user's data in a file permanently (or for the specified time). Cookie has its expiry date and time and gets removed automatically when it gets expired. Django provides in-built methods to set and fetch cookies.

**18. What is a middleware in Django?**

A middleware is a layer in Django's Request/Response processing pipeline. Each middleware is responsible for performing some specific functions on the request and/or response, such as caching, gzipping, etc.

**19. What is a QuerySet in Django?**

A QuerySet in Django is basically a collection of objects from our database. QuerySets are used by the Django ORM. When we use our models to get a single record or a group of records from the database, they are returned as QuerySets.

**20. What is Django REST Framework?**

Django REST Framework (DRF) is a Django app and a framework that lets us create RESTful APIs rapidly. DRF is especially useful if we have an existing Django web application and we wish to quickly generate an API for it.

**21. What is a context in Django?**

A context in Django is a dictionary, in which keys represent variable names and values represent their values. This dictionary (context) is passed to the template which then uses the variables to output the dynamic content.

**22. List some of the caching strategies that Django supports.**

Django supports the following caching strategies:
1. File System Caching
2. Memcached
3. In-memory Caching
4. Database Caching

**23. What is a Meta Class in Django?**

A Meta class is simply an inner class that provides metadata about the outer class in Django. It defines such things as available permissions, associated database table name, singular and plural versions of the name, etc.

**24. When should we generate and apply migrations in a Django project and why?**

We should do that whenever we create or make changes to the models of one of the apps in our project. This is because a migration is used to make changes to the database schema, and it is generated based on our models.

**25. What is serialization in Django?**

Serialization is the process of converting Django models into other formats such as XML, JSON, etc.

**26. What are generic views?**

When building a web application there are certain kind of views that we build again and again, such as a view that displays all records in the database (e.g., displaying all books in the books table), etc. These kinds of views perform the same functions and lead to repeated code. To solve this issue, Django uses class-based generic views. When using generic views, all we have to do is inherit the desired class from django.views.generic module and provide some information like model, context_object_name, etc.

**27. Which companies use Django?**

Instagram, Disqus, Mozilla, Bitbucket, Spotify, NASA, Eventbrite, etc.