**✅ Django Framework Interview Questions (Complete & No Gaps)**

**🔹 1. Django Basics**

* What is Django? Why is it popular?
* What is the architecture of Django (MVT)?
* How is MVT different from MVC?
* What is the role of manage.py?
* What is inside settings.py?
* What is the difference between a Django project and a Django app?
* How do you start a new Django project and a new app?

**🔹 2. URL Routing and Views**

* What is Django's URL dispatcher and how does it work?
* How do you map URLs using urls.py?
* What is the difference between path() and re\_path()?
* Function-based view vs Class-based view?
* What is the use of HttpResponse, render(), and redirect()?
* How does template rendering work in Django?

**🔹 3. Django Templates**

* What are Django templates and how do you use variables, filters, and tags?
* What is template inheritance?
* What is the context dictionary?
* How are static files (CSS/JS/images) handled?
* What are {% block %}, {% extends %}, {% include %} used for?

**🔹 4. Models and ORM**

* What is Django ORM? How does it work?
* Common field types: CharField, IntegerField, DateTimeField, ForeignKey, etc.
* What is the difference between makemigrations and migrate?
* What is a QuerySet?
* Common ORM queries:  
  .all(), .filter(), .get(), .exclude(), .values(), .annotate(), .aggregate()
* Model relationships:  
  OneToOneField vs ForeignKey vs ManyToManyField
* What is the use of Meta class?
* How to define a custom model manager?

**🔹 5. Forms**

* Difference between Form and ModelForm?
* How does Django form validation work?
* What are clean() and clean\_<fieldname>() methods?
* How to render and submit a form in a template?
* What is CSRF and how is it protected?

**🔹 6. Django Admin**

* How to register models in the Django admin panel?
* What is the use of ModelAdmin?
* How do you customize admin with list\_display, search\_fields, list\_filter, etc.?
* How to add custom actions in admin?
* How to create superuser?

**🔹 7. Authentication & Authorization**

* How does Django handle authentication?
* How to implement login/logout?
* What is @login\_required and @permission\_required?
* How do you create a custom user model?
* What is the difference between session and token-based authentication?

**🔹 8. Middleware**

* What is middleware in Django?
* How to write a custom middleware?
* Example: Logging time for every request and response.

**🔹 9. Django REST Framework (If Known)**

* What is Django REST Framework (DRF)?
* What are serializers? Difference between Serializer and ModelSerializer?
* APIView vs GenericAPIView vs ViewSet?
* How to build CRUD APIs using DRF?
* What is JWT authentication?

**🔹 10. Deployment**

* How do you deploy a Django project to production?
* Common platforms: Heroku, PythonAnywhere, AWS?
* What is the difference between WSGI and ASGI?
* How do you handle static and media files in production?

**🔹 11. Testing in Django**

* How do you write unit tests in Django?
* What is the difference between TestCase and SimpleTestCase?
* How to simulate requests using Django’s test client?

**🔹 12. Signals**

* What are Django signals?
* What is the use of pre\_save and post\_save?
* How to create and connect a custom signal?

**🔹 13. Caching**

* What caching backends are supported by Django?
* What is template fragment caching?
* How to use per-view or low-level caching?

**🔹 14. Security Features**

* What is CSRF and how is it prevented?
* How does Django handle XSS, SQL injection, clickjacking?
* What are ALLOWED\_HOSTS?
* How to use HTTPS and secure cookies?

**🔹 15. Real-world Scenarios**

* How do you upload and serve media files (like user profile pictures)?
* How to build a basic blog system with comments?
* How to build a shopping cart or order system?
* How to implement a dashboard with user-specific data?

**🔹 16. Optimization & Best Practices**

* What is the N+1 query problem and how to avoid it?
* What is the use of select\_related() and prefetch\_related()?
* How to add pagination in Django?
* How to add logging in Django?

**✅ Most Common Django Coding Tasks in Interviews**

* Create a blog app with post & comments using Django models & templates.
* CRUD operations with Django views and templates.
* Build a registration and login system.
* Create a REST API for TODO app using DRF.
* Upload a profile image and serve it properly.
* Add search & filter functionality to a list view.
* Use annotate() for building dashboards (ex: user count, order total, etc.)