62 Django interview questions to ask your candidates

Questions

1. Can you explain what Django is and why you would use it over other frameworks?

2. Describe the architecture of a typical Django application.

3. How does Django's ORM work and what are its advantages?

4. What are Django’s middleware and how do they function?

5. How do you manage static files in a Django project?

6. Can you explain the purpose and use of Django’s settings.py file?

7. What are Django signals and how do you use them?

8. How would you implement user authentication in Django?

9. Describe how you would optimize a Django application for better performance.

10. What steps would you take to secure a Django application?

11. Can you explain the concept of Django's MTV architecture?

12. How would you handle form validation in Django?

13. What are Django migrations and why are they important?

14. How would you optimize database queries in a Django application?

15. Explain the purpose of Django's middleware and give an example of when you might create a custom middleware.

16. How do you ensure security in a Django application?

17. Describe a challenging problem you've faced in a Django project and how you solved it.

18. What is the difference between a Django model and a database table?

19. How do you handle file uploads in Django?

20. Can you explain the use of Django's management commands and provide an example?

21. How do you implement pagination in a Django application?

22. Describe how you would manage different environments (development, staging, production) in a Django project.

23. What is Django Rest Framework (DRF) and when would you use it?

24. How do you handle internationalization (i18n) in a Django project?

25. What are class-based views and how do they differ from function-based views in Django?

26. Explain the use of Django's QuerySets and how they can be chained.

27. How do you use Django's built-in user model to extend user profiles?

28. What is the purpose of Django's 'apps' structure and how do you organize multiple apps within a project?

29. How do you integrate third-party libraries or APIs into a Django project?

30. What are custom template tags and filters in Django, and how do you create them?

31. Explain how Django handles user sessions and cookies.

32. How do you perform unit testing and integration testing in Django?

33. Can you explain the purpose of Django's URL dispatcher and how it works?

34. What are Django forms and how do they differ from HTML forms?

35. How does Django's template inheritance work, and why is it useful?

36. What is the purpose of Django's context processors?

37. Explain the concept of Django's generic views and when you would use them.

38. How does Django handle database migrations, and why are they important?

39. What is the purpose of Django's admin interface, and how would you customize it?

40. How does Django's caching system work, and when would you use it?

41. Can you explain how Django's ORM maps models to database tables?

42. How would you perform a complex query using Django's QuerySet API?

43. What is the difference between s̀elect_related̀ and p̀refetch_related̀ in Django's ORM?

44. How do you create a custom model manager in Django?

45. Can you explain the purpose of Django's g̀et_or_creatè method and provide an example of its usage?

46. How do you perform bulk operations (inserts, updates) using Django's ORM?

47. What are the implications of using Django's s̀avè method with the ùpdate_fields̀ parameter?

48. How do you handle many-to-many relationships in Django models?

49. Can you explain how to use Django's ORM for database transactions?

50. What is the purpose of the M̀età class in Django models, and how would you use it?

51. How do you handle database migrations in a multi-developer environment using Django?

52. Explain how you would use Django's ORM to filter and order query results based on specific criteria.

53. How would you approach debugging a complex issue in a Django application? Can you describe your process?

54. Imagine you are tasked with implementing a new feature that requires integrating a third-party API. How would you go about it?

55. A client reports that their Django application is running slowly. What steps would you take to identify and fix the performance issues?

56. How would you handle version control and collaboration in a Django project with multiple developers contributing code?

57. Can you describe a scenario where you had to make a trade-off between adding a feature and maintaining application performance?

58. If you had to migrate a large legacy application to Django, what would be your strategy to ensure a smooth transition?

59. Suppose you need to maintain two different versions of a Django application. How would you manage this without causing disruptions?

60. You discover a security vulnerability in your Django app. What immediate actions would you take to mitigate the risk?

61. How would you implement a caching strategy in a Django project to improve response times for users?

62. Imagine your team is adopting Django Rest Framework for a project. What best practices would you suggest to ensure a clean and maintainable codebase?