

Section 1: Django (45 Questions)

Basics (10)

1. What is Django and what problem does it solve?
2. Explain Django's MVT architecture.
3. What are Django apps? Can a project have multiple apps?
4. How do you configure a database in Django?
5. How are URLs mapped to views?
6. What is the role of `settings.py`?
7. What are middleware in Django?
8. What is the use of `INSTALLED_APPS`?
9. What's the difference between `runserver` and `startapp`?
10. What is the purpose of `manage.py`?

Models & ORM (10)

11. How do you define a model in Django?
12. What are model fields and validators?
13. How do you create and apply migrations?
14. What is a queryset?
15. How does `filter()` differ from `get()`?
16. What is `related_name` and `related_query_name`?
17. What are `ForeignKey`, `OneToOneField`, and `ManyToManyField`?
18. How can you perform raw SQL queries in Django?
19. How can you order results using ORM?
20. How do you perform a case-insensitive search?

Views & Templates (10)

21. What are function-based views vs. class-based views?
22. What are mixins in CBVs?
23. How do you pass context to templates?

24. What is template inheritance?
25. What are template filters and how to use them?
26. How do you use loops and if statements in Django templates?
27. How do you handle 404 errors in Django?
28. What is the `render ()` function?
29. How do you use static files in Django?
30. How do you write a custom template tag?

Forms & Admin (10)

31. What is the difference between `Form` and `ModelForm`?
32. How do you validate form inputs?
33. How do you customize the Django admin interface?
34. How do you register models in admin?
35. How to add search and filters in admin?
36. What is CSRF and how does Django handle it?
37. How do you use messages framework in Django?
38. How can you implement file uploads?
39. How do you implement login/logout?
40. How do you use `@login_required` decorator?

Advanced Django (5)

41. What is the signals framework?
42. How do you implement pagination?
43. How to optimize Django queries?
44. How do you handle background tasks in Django?
45. How to deploy a Django project on production?

Section 2: Django REST Framework (25 Questions)

46. What is Django REST Framework (DRF)?
47. What are serializers? What is `ModelSerializer`?

48. How do you create an API view in DRF?
49. What is the difference between `APIView` and `ViewSet`?
50. What are routers in DRF?
51. How to perform CRUD operations using DRF?
52. How do you handle validation errors in serializers?
53. What is the purpose of `@api_view`?
54. How do you implement Token Authentication?
55. How do you secure your API using permissions?
56. What is `IsAuthenticated` vs. `AllowAny`?
57. What are throttling and rate limits?
58. How to upload files via an API?
59. What is `HyperlinkedModelSerializer`?
60. How can you filter results in DRF?
61. How to handle nested relationships in serializers?
62. What are generic views in DRF?
63. How to return custom JSON responses?
64. How to implement pagination in DRF?
65. How to test your API endpoints?
66. How to create versioned APIs?
67. How to return HTTP 400/404/500 errors?
68. What is the difference between PUT and PATCH?
69. How do you serialize a queryset with nested models?
70. How do you use Django filters with DRF?

Section 3: SQL (30 Questions)

Basic SQL (15)

71. What is a primary key?
72. What is a foreign key?

73. What are **SELECT**, **WHERE**, **ORDER BY**, **GROUP BY** used for?
74. Write a query to fetch all users older than 18.
75. Write a query to get the second highest salary.
76. What is the difference between **WHERE** and **HAVING**?
77. How to update a specific row in a table?
78. How to delete duplicate rows?
79. What is a **NULL** value?
80. How do you count the number of users per city?

Joins (10)

81. What is a **JOIN** in SQL?
82. What is the difference between **INNER JOIN** and **LEFT JOIN**?
83. What is a **FULL OUTER JOIN**?
84. Write a query using **INNER JOIN** between **users** and **orders**.
85. What happens when you **JOIN** on non-indexed columns?
86. How to perform self join?
87. How do you join more than 2 tables?
88. What is a **CROSS JOIN**?
89. What is the purpose of **ON** clause?
90. How to handle **NULLs** during **JOIN**?

Advanced SQL (5)

91. What are window functions?
92. Write a query using **RANK ()** or **ROW_NUMBER ()**.
93. What is a CTE (Common Table Expression)?
94. What is indexing and how does it improve performance?
95. How to optimize large queries?

Section 4: JavaScript (30 Questions)

Basics (10)

96. What is the difference between `var`, `let`, and `const`?
97. What are data types in JS?
98. What are arrays and objects?
99. How do you define and call a function?
100. What is a callback function?
101. What is the use of `this` keyword?
102. What is hoisting?
103. What are arrow functions?
104. How do you use `map()`, `filter()`, `reduce()`?
105. What is the difference between `==` and `===`?

DOM & Events (10)

106. What is the DOM?
107. How do you select an element using JS?
108. How do you add a click event to a button?
109. How do you change HTML content using JS?
110. What is event bubbling?
111. How to toggle a CSS class using JS?
112. What is `addEventListener()`?
113. How to get form input values?
114. What is the use of `preventDefault()`?
115. How do you validate a form in JS?

Fetch API & Async (10)

116. What is the `fetch()` API?
117. How do you handle JSON responses?
118. What is `async/await` in JS?
119. How do you make an API call using `fetch()`?

- 120. What is promise chaining?
- 121. What is the difference between `async/await` and promises?
- 122. How to handle API errors in JS?
- 123. Write a function to fetch and display data from an API.
- 124. What is CORS and how does it affect API calls?
- 125. How do you send POST data using `fetch()`?

Section 5: REST API Concepts (10 Questions)

- 126. What does REST stand for?
- 127. What makes an API RESTful?
- 128. What are the common HTTP methods used in REST?
- 129. What are status codes: 200, 201, 400, 401, 404, 500?
- 130. What is the difference between PUT and PATCH?
- 131. What are headers in an HTTP request?
- 132. What is the role of authentication in APIs?
- 133. How does a client consume an API?
- 134. What is JSON and why is it used in REST APIs?
- 135. What are idempotent methods in REST?

Section 6: Debugging, Logic & Aptitude Coding (10 Questions)

- 136. Fix this Django bug: `FieldError: Unknown field(s) (xyz) specified for Note`
- 137. A view returns empty JSON. What might be the issue?
- 138. Why is your query running slow? How would you fix it?
- 139. Write a Python function to check if a string is a palindrome.
- 140. Write JS code to reverse a number.
- 141. You deployed a Django API, but the POST request fails. What's your checklist?
- 142. Write a SQL query to get the latest N records from a table.
- 143. How to implement rate-limiting in a Django API?

- 144. How would you debug a JavaScript function that gives `undefined`?
- 145. How to test a REST API locally without a frontend?

Section 7: Behavioral & Project Questions (5 Questions)

- 146. Tell us about a project you built using Django.
- 147. What was the most challenging bug you solved?
- 148. How do you handle tight deadlines or pressure?
- 149. Why do you want to work at Botxbyte?
- 150. What are your long-term goals as a developer?

Section 8: Advanced Django & DRF (20 Questions)

- 151. What is lazy loading in Django ORM?
- 152. How do you handle circular imports in Django?
- 153. What are `select_related` and `prefetch_related`?
- 154. Explain how Django handles transactions.
- 155. How to log SQL queries in Django?
- 156. What is `@receiver` in Django signals?
- 157. How to create a custom Django management command?
- 158. How to implement user roles/permissions beyond the default?
- 159. How do you override the `save()` method in models and why?
- 160. How to integrate third-party APIs in Django views?
- 161. How to build a nested serializer in DRF?
- 162. What's the difference between `Serializer` and `ModelSerializer`?
- 163. How to validate a single field in a DRF serializer?
- 164. How to write custom error messages in DRF?
- 165. How does DRF handle file/image uploads?
- 166. What's the difference between throttling and rate limiting?
- 167. How do you test DRF APIs using Postman or Curl?
- 168. Explain the flow of request to response in DRF.

169. How do you return paginated API responses with metadata?

170. How to integrate JWT authentication in a DRF app?

Section 9: SQL Deep Dive (10 Questions)

171. What is the difference between UNION and UNION ALL?

172. Write a query to get the 3rd and 4th highest salaries from an employee table.

173. How do you detect and resolve duplicate records?

174. What is a stored procedure? When would you use it?

175. What are views in SQL? What are their pros and cons?

176. What is indexing? When can it cause performance *degradation*?

177. How do you handle NULLs in calculations?

178. What's the use of COALESCE () in SQL?

179. Explain a case where GROUP BY + HAVING improved a report's performance.

180. What's the difference between TRUNCATE, DELETE, and DROP?

Section 10: Advanced JavaScript & Browser (10 Questions)

181. What is a closure in JavaScript? Provide an example.

182. What is event bubbling vs. capturing?

183. What is the call stack in JS?

184. What is debouncing and throttling in JS?

185. What is the difference between synchronous and asynchronous code?

186. What is the use of `localStorage`, `sessionStorage`, and `cookies`?

187. How to deep clone an object in JS?

188. What are promises? How is `.then()` different from `async/await`?

189. Explain hoisting with an example.

190. What is `try...catch` used for in JS?

Section 11: REST API + Security & System Design Basics (10 Questions)

191. What are the common security issues in REST APIs and how to prevent them?
192. What is rate limiting? How would you implement it?
193. What is the difference between stateless and stateful APIs?
194. How do you structure a RESTful URL for nested resources?
195. Explain how authentication differs from authorization in REST APIs.
196. How do you prevent brute-force login attempts in Django?
197. What is idempotency and which HTTP methods are idempotent?
198. What is the difference between HTTP and HTTPS?
199. How would you design an API to support pagination and filters?
200. What are webhooks and how are they different from APIs?