INTERVIEW QUESTIONS

I've listed below few questions that I can remember, these are from multiple companies

(i gave interviews for django mostly, so there aren’t many flask related questions here)

1. OOPs principles

2. Difference between list and tuples, when to use tuple over list

3. Difference between array and list

4. what is iterator, generator

5. What are high order functions

6. What are decorators

7. What are generic functions

8. How to remove redundant values from a list whose length is more than 10k, without using set, and without creating another list as it is not optimal solution. tell complexity for your solution.

9. What's pickling

10. What's namespace, it's significance

11. Diff bwtn cloning, shallow copy and deep copy

12. Have u used random in project, where?

13. what are loggers, where did u use that

14. If you haven't written a logging mechanism in the application and now all the coding is done and appl is ready to deploy how would you integrate the logging in this scenario

    ****use decorators to write logging mechanism

15. @func2

```
@fun3
def fun1():
    pass
```

    Explain above execution flow

16. difference bwtn multiple and multilevel inheritance. Is multiple inheritance possible in python? where was it used -->***Mixin classes

17. what's init and new

18. explain mechanism of reduce in lambda

19. write git command to create new branch
20. diff bwtn rebase and merge

21. how did u review code, was the code stored on github for review?

22. your code is complete and on your machine but there is still time to push on github, so what are your activities in this time

23. name the libraries you've frequently used

24. what's the most challenging thing you've faced in the project and how did u overcome that
25. how are exceptions written in code handled, are they directly passed to user? or is there any other mechanism that you've followed.

26. what did u use github for

27. what version control tool did u use?

28. your role in the project

29. architecture of your project, is it a micro service, monolithic or any other ?

30. diff bwtn django and flask and which one to use when.

31. what's related_name in django, it's significance

32. names of default tables, middlewares, installed apps in django

33. explain django architecture.

34. explain request and response flow in django

35. what are models their significance

36. significance of settings.py file in django

37. orms used in flask and django

38. attributes of render()

39. format of request and response (syntax)

40. explain authentication and authorization

41. project directory structure of django

42. how to use file-based sessions?

43. command to get all models in django

44. what are session and cookies, their significance

45. what's cache, was that used in your project, and for what purpose

46. explain the purpose of middlewares, which custom middleware have you developed in the project?

47. explain MVT

48. can you get static files from your project, how?

49. what's blueprint in flask and was that used in your project?

50. Can u iterate the objects of a class in python like we do using orm in django and flask and how? → ** it is done using generator/iterator(I don’t remember exactly)

51. q = Employee.objects.filter(‘sub’ = ‘comp’)

```
 q = q.filter(‘marks’ >50)
q = q.sort
 print(q)
```

In above how many times the database was invoked?

52. difference between get and filter

53. basic django commands and queries

54. use of select_related and prefetch_related

55. can you get data without using select_related and how?

56. the queries written using orms are lazy, why?

57. what’s Abstract and why to use that

58. write sample example for inheritance

59. diff bwtn class method and static method

60. how to define relationship between tables using orm

61. total number of joins -> inner, outer, left, right, etc.

62. diff bwtn primary key, unique key, foreign key

63. is Null present in unique and primary columns?

64. diff bwtn delete, truncate, drop

65. why do we use left join, what would be output of following after left join

T1	T2
C1	C2
1	115
2	15
6	6
15	20
49	59

66. diff bwtn sql and non sql dbs

67. can we have joins in non sql dbs?

68. What is Pagination and write sample code for that

69. what’s REST API, it’s significance

70. was your project/api scalable, to what extent and how?

71. what performance testing was done in the project?

72. team size, no. of backend dev, frontend dev(were they committed to frontend development only), were team leads involved in development?

73. which clauses have you used in the project → eg. Where, between, etc.

74. who are the users of your application, is it live ?

75. have you implemented a project/module from scratch?

76. were you involved in scrum or sprint planning

77. have you ever told the estimated time reqd for modules during sprint

78. are u a team player or an individual contributor?

79. have u suggested any changes in the development to optimize the code

80. on an average, how many lines of code do you write in a day?

81. diff bwtn module and package

82. how can a user(or how would u know that the user is auhorized) use db to add any data since the dbs are secured

83. diff mechanisms of authentication → authentication, token based, etc

85. how are apis secured?

86. diff bwtn http and https

87. In your application suppose there are 3 pages, login, register, dashboard, so only after login user should be taken to dashboard otherwise not.

XYZ person is using this application frequently and knows most of the urls. So, before logging in to the app he is hitting dakhboard page, but Is redirected to login page. How this happens, how you implement such mechanisms? → using session

88. what is access token?

89. what's is csrf token and it's significance

90. can you use multiple dbs in a django project, how?

91. what are serializers?

92. what are loggers, write a sample code for logging

93. tell diff status codes and what they represents →100x, 200x,etc

94. different methods used in apis → get, post, put, delete, etc.

95. what is jsonify, why to use them when we can convert data from python to json and vice versa without them as well?

96. how to do multithreading in python

97. what are model managers, where are they used?

98. where was ur project deployed?

99. how to move data from one db to another(eg. Mysql to postgres)

100. how to convert python obj to json ad vice versa

101. diff types of viewsets in django and their difference

102. signals in django, where were they used, are those decorators?

103. types of design patterns and name those which you've used?

104. what are series, dataframe in numpy

105. have u used numpy/pandas in your project, if yes, where?

106. have you automated anything in the project(since I had selenium in my resume)

107. types of arguments

108. did you write unit test cases for your module by yourself, or was that someone else’s task to do?

109. use of clone in git

110 . how to commit, push, pull in git/github

FEW CODING QUESTIONS ASKED

Most of the answers are present on below sites :

https://www.geeksforgeeks.org/python-programming-examples/

OR

https://www.programiz.com/python-programming/examples

1. input = [1,1,5,7,7,1]

   output = {1 : 3, 5 : 1, 7 : 2 } --> basically count of each number in a list

   or

   input = 'mississippi'

   output = {'m' :1, 'i' : 4, 's' : 4, 'p' : 2 }

2. given a list of integers, find the second largest value in the list

   eg, input = [12, 34, 10, 67, 12, 96]

   output = 67

3. find unique pairs of numbers from list whose sum is equal to k

   eg, input = [1,3,4,7,5,6]

   k = 10

output = [(3,7), (4,6)]

4. a = [3,5,7,9,13]

   b = [4,6,10,12,60]

   write a function to sort get

   output = [3,4,5,6,7,9,10,12,13,60]

```
def sorted(a,b)
    ---write your logic here---
```

***hint : you can use merge sort logic as lists a and b are already sorted in themselves.

**make sure you know the complexity of bubble sort and merge sort

(they ask u the comlexity of code that u've written)

5. input = 'aaabbbbccccddaa'

   output = 'a3b4c4d2a2'

   or vice versa

6. input = [101100 011100]

   output = [010011100011]

7. write a decorator function

8. print Jane's marks

```
mydict = {
                    'clg' : {
                              'cls': {
```

```
                    'div' : 'a',
                    'records' : {
                                1 :{'name' : 'Dane', 'marks' : 98},
                                2 :{'name' : 'Jane', 'marks' : 68}
                                }
                        }
                    }
        }
```

***make sure you know the inbuilt functions of dictionary

9. change Joy's marks to 368

```
mydict = {
        1 : {'name' : 'Dane', 'phy' : '200', 'maths' : 100,},
        2 : {'name' : 'Jane', 'phy' : '300', 'maths' : 200,},
        3 : {'name' : 'Joy', 'phy' : '200', 'maths' : 300,}
        }
```

10. mydict = {'a' : 100, 'b' : 200, 'c' : 300}

    take a number from user and return True if it is present in mydict, else return False

11. create a list from [1,2,3,4,5,6,7,8,9] with all even numbers

    - using loop
    - using list comprehensions
    - using lambda function

12. merge two dictionaries --***hint : use update()

input --> d1 = {1:2}, d2 = {2:1}

output --> {1:2, 2:1}

13. input = 'i live in MH'

output = 'MH in live i'

14. what's lamda, it's significance, lambda function to write a a+b*c and print output of the same with any numbers of ur choice

15. what are generators, their significance, sample code for generator

16. what are decorators. write decorator for function fun()

input -->

```
def fun():
    print("Hi")
```

output -->

1. write decorator to print "Hello" instead of "Hi"

2. print both "Hello" and "Hi"

17. write a django application for parking , where user can view the empty parking slots online, book one and once the car moves out from the slot the number of empty slots should get updated

18. write a sample flask application

19. input --> url : someurl with lots of json data

output --> print the data on console -->(requests library)

20. write code to obtain fibonacci series

21. import random

```
my_list = list(range(0,101))

random_item_from_list = random.choice(my_list)

my_list.remove(random_item_from_list)

random.shuffle(my_list)

print(my_list)
```

Q --> print(random_item_from_list) -->write code to print this number

***hint : take sume of all numbers in list and subtract with original list's sum

22. a = 10 , b = 20. Swap the numbers in multiple ways --> using 3rd var, without 3rd variable

solution :

```
temp = a
a = b
b = temp
OR
a,b = b,a
```

23. write a code to reverse the string without using built-in functions

24. Q1 . write a class to print "Hello" in constructor

Q2 . write a child class from above cls and print "Hi" in it's constructor

Q3. print both "Hello" and "Hi" written in constructor of above classes

25. what’s *args and **kwargs

```
def fun(*a, **b):
    print(a,b)
fun(1, 2, 3, a=10, b=20)
```

what would be output for a and b?

26. list1 =[1, 2, 3, ……. 25]

Interval = 1

Write a code to get a list with given interval and make sure 1 and 25 are always present in output list no matter what the interval is.

After writing a code, interval was changed to 1.5 , 2 to see if it gives correct output

There were scenario based questions as well, few I have mentioned and rest I don’t remember(sorry!).

There were few questions based on advanced concepts of django (version 3.x) but then the interviewer asked the version of django that I’ve used (it was 2.x) so he explained few things of version 3 that are not in 2.

I hope these questions help you guys in the interviews.

ALL THE BEST!