Viva

Python - General Questions

1. What are the key differences between Python 2.x and Python 3.x?
2. What are decorators in Python, and how do they work?
3. Explain the difference between a list and a tuple in Python.
4. How does Python handle memory management?
5. What are Python's built-in data types?
6. What are list comprehensions? Can you give an example?
7. What is the purpose of the `self` keyword in Python classes?
8. What is the difference between `deepcopy` and `shallow copy`?
9. How does Python handle exception handling? Give an example.
10. What is a lambda function in Python, and how does it differ from a regular function?
11. Explain the purpose and use of the `with` statement in Python.
12. What are modules and packages in Python? How are they used?
13. How can you handle errors using Python's `try`, `except`, `finally`?
14. Explain the concept of Python's `Global Interpreter Lock (GIL)`.
15. How do you manage dependencies in a Python project?
16. What are generators in Python, and how do they differ from regular functions?
17. Can you explain the difference between `is` and `==` in Python?
18. How do you handle multithreading and multiprocessing in Python?
19. What are Python's built-in functions for iterating over data?
20. What are Python decorators used for, and how do they enhance functionality?

Flask - Framework Specific Questions

1. What is Flask, and how does it differ from other web frameworks like Django?
2. What is the role of the `Flask` class in a Flask application?
3. What is Flask's routing system? How do you define routes?
4. Explain the purpose of `render_template` in Flask.
5. How do you handle static files in Flask (CSS, JS, images)?
6. What is the purpose of the `url_for` function in Flask?
7. How do you handle form data in Flask?

8. Explain Flask’s request and response objects.
9. What are Flask blueprints, and how do they help in structuring a Flask application?
10. How do you set up sessions in Flask?
11. What is Flask’s `flash` method, and how do you use it?
12. Explain the `before_request` and `after_request` decorators in Flask.
13. What is the difference between Flask and Flask-RESTful?
14. How do you handle HTTP methods (GET, POST, PUT, DELETE) in Flask?
15. What are Flask extensions, and can you give examples?
16. How can you handle user authentication and authorization in Flask?
17. How do you integrate Flask with a database (e.g., SQLAlchemy)?
18. What is the role of middleware in Flask?
19. How do you handle file uploads in Flask?
20. How do you test a Flask application?

HTML & CSS - Frontend Questions

1. What is the difference between HTML5 and previous versions of HTML?
2. What is the difference between `id` and `class` attributes in HTML?
3. How do you embed an image in HTML?
4. What is the purpose of the `<meta>` tag in HTML?
5. Explain the difference between inline and block-level elements in HTML.
6. What is the role of semantic HTML, and why is it important for accessibility?
7. What are the new elements introduced in HTML5?
8. How do you link an external stylesheet to an HTML file?
9. How do you create a table in HTML? Explain its structure.
10. What are forms in HTML, and how do you handle user inputs?
11. What is the `box-model` in CSS, and how does it affect layout?
12. What are the different ways to apply CSS to an HTML document?
13. Explain the difference between relative, absolute, fixed, and sticky positioning in CSS.
14. What is the purpose of the `z-index` property in CSS?
15. How can you center an element horizontally and vertically using CSS?
16. What is the difference between `display: block` and `display: inline` in CSS?
17. How do you make a website responsive using CSS?
18. What are media queries, and how are they used in CSS?
19. What is the purpose of Flexbox in CSS, and how is it different from CSS Grid?
20. What are pseudo-classes and pseudo-elements in CSS?

JavaScript - Frontend Questions

1. What is the difference between `let`, `var`, and `const` in JavaScript?
2. How do JavaScript closures work, and what are their use cases?
3. Explain the concept of hoisting in JavaScript.
4. What are promises in JavaScript, and how do they work?
5. What is the difference between synchronous and asynchronous programming in JavaScript?
6. What are JavaScript events, and how do you handle them?
7. Explain the concept of callbacks in JavaScript.
8. What is the difference between `==` and `===` in JavaScript?
9. What is the role of the `this` keyword in JavaScript?
10. How do you handle errors in JavaScript?
11. What are JavaScript arrow functions, and how do they differ from traditional functions?
12. What is the DOM, and how do you manipulate it using JavaScript?
13. How do you make an AJAX call in JavaScript?
14. What are the main differences between `null` and `undefined` in JavaScript?
15. What is event delegation in JavaScript, and how is it used?
16. What are higher-order functions in JavaScript? Can you give an example?
17. What are closures in JavaScript, and how do they work?
18. Explain the difference between `apply()`, `call()`, and `bind()` in JavaScript.
19. What are JavaScript modules, and how do you import and export them?
20. What is the concept of prototypal inheritance in JavaScript?

Full-Stack Development (Integration) Questions

1. What is the role of Flask in a full-stack web application?
2. How do you connect the frontend (HTML, CSS, JS) with the backend (Flask)?
3. How do you use AJAX to send data from the frontend to Flask (backend)?
4. What are REST APIs, and how do you implement them using Flask?
5. How do you handle CORS (Cross-Origin Resource Sharing) in Flask?
6. How do you manage state in a full-stack application?
7. How can you make a Flask application secure (e.g., SQL Injection, CSRF)?
8. Explain how you would implement a login system with JWT tokens in Flask.
9. How do you deploy a full-stack Flask application to a production server?

10. What are the common best practices for structuring a Flask-based full-stack application?
11. How do you use Python's `requests` library to make API calls from Flask to another service?
12. What is the importance of templating in Flask and how does it help integrate with HTML?
13. How can you optimize the performance of a Flask application?
14. What are some ways to scale a Flask application?
15. How would you add error handling and logging in Flask?
16. What is a session in Flask, and how would you store user authentication information?
17. How do you integrate Flask with a frontend framework like React or Angular?
18. How can you implement file upload functionality in a Flask-based full-stack application?
19. How would you connect a Flask app to a database like PostgreSQL or MySQL?
20. What tools or techniques would you use to debug a full-stack application?