1. Explain the event loop in Node.js?
2. What is the difference between promises and callbacks?
3. How do you structure a large Node.js application?
4. How do you spawn child processes in Node.js?
5. What is middleware and how do you use it in Express.js?
6. How does Node.js handle asynchronous operations?
7. Explain the event-driven, non-blocking I/O model in Node.js.
8. What is the difference between require and import?
9. How do you manage dependencies in a Node.js project?
10. What is the purpose of package json?
11. How do you handle errors in asynchronous operations in Node.js?
12. What is the difference between synchronous and asynchronous error handling?
13. How do you set up routing in an Express application?
14. Explain the difference between application-level and router-level middleware.
15. How do you implement authentication and authorization in Express.js?
16. How do you handle authorization in an Express app?
17. How would you write custom error-handling middleware in Express?
18. What are some common security practices in Express?
19. How do you prevent SQL injection and Cross-Site Scripting (XSS) in an Express application?
20. How do you perform CRUD operations in MongoDB?
21. What is the difference between embedded documents and references in MongoDB?
22. When and how do you use indexes in MongoDB?
23. Explain how to use the aggregation framework in MongoDB.
24. How do you define a schema in Mongoose?
25. Explain the MongoDB document structure.
26. How do you model relationships between data in MongoDB?
27. How do you handle different content types in APIs?
28. How do you implement pagination in APIs?
29. How do you handle rate limiting in APIs?
30. How do you test your backend code?
31. How do you deploy a Node.js application?
32. How do you handle database migrations?
33. How do you approach debugging a Node.js application?
34. What are pre and post hooks in Mongoose, and how do you use them?
35. How do you handle multi-document transactions in MongoDB?
36. What are RESTful APIs, and how do you design one?
37. How do you handle versioning in an API?
38. How do RESTful APIs differ from GraphQL?
39. What are some best practices for designing RESTful APIs?
40. What is the difference between NoSQL and SQL databases?
41. When would you choose MongoDB over a SQL database?
42. How do you perform backup and restore operations in MongoDB?
43. How do you optimize database queries in MongoDB?
44. How do you write unit tests for a Node.js application?
45. What testing frameworks have you used, and why?
46. How do you perform integration testing in a MERN application?
47. What is test coverage, and why is it important?
48. How do you measure and improve test coverage?
49. How do you implement caching in a Node.js application?
50. What strategies do you use to optimize database performance?
51. How do you handle load balancing in a Node.js application?
52. What are the key considerations for scaling a MERN stack application?
53. How do you manage environment variables in a Node.js application?
54. What is your experience with CI/CD pipelines?
55. How do you deploy a Node.js application to AWS or Heroku?
56. How do you ensure your application is production-ready?
57. How do you validate and sanitize user input in a Node.js application?
58. How do you encrypt sensitive data in a MERN application?
59. What steps do you take to prevent CSRF attacks?
60. How do you secure your API endpoints in an Express application?

**LINDA**

1. What is node.js?
2. How can you avoid callback hell?
3. When are background or worders processes useful?
4. Why is Node.js single threaded?
5. Name the types of API functions in node.js?
6. Explain node.js in chaining?
7. What are streams in node.js and explain the different types of streams present in node.js?
8. What is package json?
9. Explain the purpose of module export
10. List down the different security implementation within node.js
11. Explain the concept of URL model
12. What is middleware?
13. Explain libuv?
14. List down the two arguments that async queue takes as input?
15. Differentiate between the spawn and fork method in node.js
16. Explain the purpose of the express package
17. Explain the buffer classes in the node
18. How does node handle the child threads
19. Explain the stream in node along with its various types
20. Describe the exist code in node
21. Is cryptography supported in node
22. Explain the reason as to why express app and server folder must be kept separate
23. What is the role of asset module folder in node
24. What is the role of async\_hook module in node
25. What are the buffer objects in node
26. What are the different ways of implementing Addons in node
27. How can we spawn the child process asynchronously without blocking the node event loop
28. How can we take advantages of multi core system in node as node works on single thread
29. What is datatype of the console
30. Which are the different console methods available
31. Can node perform cryptography function
32. How can we read and write file in node
33. Which are the global objects in node
34. How can we perform asynchronous network APIs in node
35. What are the utilities of OS module in node
36. Which are the areas where it is suitable to use node
37. Which are the areas where it is not suitable to use node
38. What are the key features of node
39. Explain the REPL in node
40. How can you write CRUD operation in node without using framework
41. What is difference between node AJAX and JQuery
42. What is eventEmitter in node
43. What is a Child\_process module in node
44. What do you mean by asynchronous API
45. What is the benefit of using Node
46. Is node single threaded app
47. What is global installation of dependencies
48. What is local installation of dependencies
49. How do you check the already installed dependencies
50. Name some of the attributes of package json
51. How do you uninstall a dependency using npm
52. How to update dependency using npm
53. What is callback
54. What is blocking code
55. How node prevents blocking code
56. What is event loop
57. What is event emitter
58. What is the purpose of buffer class in node
59. What is piping in node
60. Which module is used for buffer-based operation
61. what is the difference between synchronous and asynchronous methods of fa module
62. what are streams
63. how many types of streams are present in node
64. name some of the event fired by stream
65. what is the first-class function in JS
66. what is node and how it works
67. how do you manage package
68. how is node better than other frameworks most popularly used
69. explain the steps how “Control Flow” control the functions calls
70. what are the some commonly used timing features of node
71. what are the advantages of using promise instead of callback
72. what is fork in node
73. why node is single threaded
74. list down the two arguments that async\_queue takes as input
75. how does node overcome the problem of blocking of I/O operations