MongoDB

1. What is MongoDB?
2. How does MongoDB differ from a relational database?
3. What are the advantages of using MongoDB?
4. Explain the concept of a collection in MongoDB.
5. What is a replica set in MongoDB?
6. What is sharding in MongoDB and why is it used?
7. How do you perform CRUD operations in MongoDB?
8. What is an index in MongoDB and why is it important?
9. Explain the difference between find and findOne methods in MongoDB.
10. How do you handle transactions in MongoDB?

Express

1. What is Express.js?
2. What are some features of Express.js?
3. How do you set up a basic Express server?
4. Explain the concept of middleware in Express.
5. How do you handle routing in Express?
6. How do you handle errors in Express?
7. What are some common middleware libraries used with Express?
8. How do you implement authentication in an Express application?
9. How do you connect an Express application to MongoDB?
10. What is the purpose of the next function in Express middleware?

React

1. What is React?
2. What are the main features of React?
3. Explain the concept of a component in React.
4. What are the differences between functional and class components?
5. What is JSX and why is it used in React?
6. Explain the concept of state and props in React.
7. What are hooks in React and why are they used?
8. What is the Context API in React?
9. How do you handle forms in React?
10. What is React Router and how do you implement it?
11. Explain the concept of lifecycle methods in React class components.
12. What is the difference between controlled and uncontrolled components?

Node.js

1. What is Node.js?
2. What are the main features of Node.js?
3. Explain the event-driven architecture of Node.js.
4. What is the purpose of the package.json file in a Node.js project?
5. How do you manage dependencies in a Node.js project?
6. What is NPM and how is it used in Node.js?
7. How do you handle asynchronous operations in Node.js?
8. What are streams in Node.js and how do you use them?
9. Explain the concept of middleware in Node.js.
10. How do you handle file uploads in Node.js?

Full-Stack / General

1. How does the MERN stack architecture work?
2. How do you manage state in a full-stack application?
3. What is the role of RESTful APIs in the MERN stack?
4. How do you secure a MERN stack application?
5. How do you deploy a MERN stack application?
6. Explain the process of authentication and authorization in a MERN stack application.
7. What are some best practices for optimizing performance in a MERN stack application?
8. How do you handle environment variables in a MERN stack application?
9. Explain the concept of server-side rendering (SSR) in React.
10. How do you test a MERN stack application?