# **✅ INTERVIEW QUESTIONS MASTER LIST (Topic-wise)**

## **🔹 1. JavaScript (Basics + ES6+)**

### **📌 Variables & Scope**

* What is the difference between var, let, and const?
* What is block scope?
* What is hoisting?

### **📌 Functions & Arrays**

* What is a higher-order function?
* What is a callback function?
* What is the difference between map() and forEach()?
* How does map() work?
* What does reduce() do?
* How does filter() work?
* What is a pure function?

### **📌 Promises & Async/Await**

* What is a Promise?
* What is async/await?
* How does async/await work internally?
* What is the difference between synchronous and asynchronous code?

### **📌 Miscellaneous Concepts**

* What is the DOM?
* What is the event loop in JavaScript?
* What is the difference between == and ===?
* What is the difference between null and undefined?
* What is a closure?
* Explain the this keyword in JavaScript.
* What are arrow functions?
* What are template literals?
* What is destructuring in JavaScript?
* What are the spread and rest operators?
* What is the output of console.log(typeof NaN);?

## **🔹 2. React.js**

### **📌 Core Concepts**

* What is React?
* What is the difference between a component and an element?
* What are functional components?
* What is JSX and how is it different from HTML?
* Can a component return multiple elements?

### **📌 Props & State**

* What is the difference between props and state?
* How does data flow in React?

### **📌 Hooks**

* What is useState and how do you use it?
* What is useEffect and how does it work?
* When does useEffect run?
* Why were hooks introduced?
* Can you use hooks inside loops?
* What happens if useState is updated in a loop?

### **📌 Events**

* How do you handle events in React?
* How do you prevent default behavior in React?

### **📌 List Rendering**

* How do you render a list in React?
* Why do we need keys in a list?

### **📌 Forms & Inputs**

* How do you manage forms in React?
* What are controlled vs uncontrolled components?
* How do you lift state up?

### **📌 Performance**

* What is the virtual DOM?
* What is React reconciliation?
* How can you optimize performance in React?

## **🔹 3. Node.js**

### **📌 Core Concepts**

* What is Node.js?
* What is the difference between require and import?
* What is the role of package.json?
* What is nodemon?
* What is the difference between synchronous and asynchronous code in Node?

### **📌 Error Handling & Status Codes**

* How do you handle errors in Express?
* What are status codes (200, 201, 400, 401, 403, 404, 500)?

## **🔹 4. Express.js**

### **📌 Basics**

* What is Express.js?
* Why do we use Express?

### **📌 Routing & Middleware**

* How do you define a route in Express?
* What is middleware in Express?
* What does express.json() do?
* What is the order of middleware execution?
* How do you create dynamic route parameters like /user/:id?

### **📌 API Security**

* What is CORS?
* How can you secure an Express API?

## **🔹 5. MongoDB + Mongoose**

### **📌 Core MongoDB**

* What is MongoDB?
* What is a collection and what is a document?
* How do you find, insert, update, and delete documents?

### **📌 Mongoose**

* What is a schema in Mongoose?
* How do you define a model in Mongoose?
* How do you connect MongoDB with Node.js?
* What is Model.findById()?
* How do you handle validation in Mongoose?
* What happens if the database is not connected properly?
* How do you define relationships in MongoDB?

## **🔹 6. REST APIs**

### **📌 REST Basics**

* What is a REST API?
* Difference between GET, POST, PUT, and DELETE?
* What is status code 201 used for?
* How do you send JSON data from client to server?
* Difference between PUT and PATCH?

### **📌 Design & Error Handling**

* Design a REST API for a blogging system (GET all posts, POST new post).
* How would you handle errors in a REST API?

## **🔹 7. Git & GitHub**

### **📌 Basics**

* What is Git and why is it used?
* What is the difference between Git and GitHub?

### **📌 Commands & Workflow**

* How do you clone a repo?
* What does git add, git commit, and git push do?
* How do you check changes before committing?
* How do you create a pull request?
* What is the difference between origin and upstream?
* What is a merge conflict and how do you resolve it?

## **🔹 8. Project-Based Questions**

### **📦 Secure Blogging Platform**

* What tech stack did you use?
* How did you implement authentication?
* How do you handle role-based access?
* What are the models in your backend?
* How does your blog page load data?
* How did you handle frontend/backend communication?
* What did you learn from this project?

### **📦 PhishGuard Pro**

* How do you detect phishing patterns?
* How did you integrate OpenAI or VirusTotal API?
* How does your Chrome extension interact with pages?
* What architecture or structure did you use?

## **🔹 9. HR / Personal Questions**

### **👔 General**

* Tell me about yourself.
* Why do you want this internship/job?
* What projects have you worked on?
* What challenges did you face during your projects?
* Are you comfortable working onsite from 10 to 7?
* Are you open to working in a team?
* Where do you see yourself in the next 1-2 years?
* What do you know about our company?
* Why should we hire you?
* How do you handle bugs or errors in your code?
* Are you comfortable learning new technologies quickly?

## **✅ Want These As:**

* 📄 **PDF version for printing or revision?**
* 💬 **50 short answers for top questions?**
* 🎯 **Mock Interview based on these?**

Just say the word and I’ll send it your way!

4o

##### **You said:**

how will i get answers of these