

INTERVIEW PREPARATION GUIDE

# TOP 80

## Full Stack Development Interview Questions & Answers

A single, structured guide covering everything from HTML & JavaScript basics to React, Node.js & APIs, databases, Java Spring Boot, and Git / DevOps — built for candidates preparing for their next full stack interview.

## CONTENTS

# What's Inside

80 interview questions, grouped into six focused segments — go through them in order, or jump straight to the topic you need to revise.

|           |                                                                                                                        |                  |
|-----------|------------------------------------------------------------------------------------------------------------------------|------------------|
| <b>01</b> | <b>Basic Full Stack Development</b><br>Foundations every full stack candidate must know                                | <b>Q1 – Q15</b>  |
| <b>02</b> | <b>Front-End and React</b><br>UI frameworks, components, and browser fundamentals                                      | <b>Q16 – Q30</b> |
| <b>03</b> | <b>Back-End and API</b><br>Servers, APIs, and how the pieces talk to each other                                        | <b>Q31 – Q45</b> |
| <b>04</b> | <b>Database</b><br>Storing, relating, and querying data reliably                                                       | <b>Q46 – Q60</b> |
| <b>05</b> | <b>Java Full Stack Developer</b><br>OOP concepts and the Java enterprise stack                                         | <b>Q61 – Q70</b> |
| <b>06</b> | <b>Git, DevOps, Security &amp; Scenarios</b><br>Version control, deployment, and how you'd handle it in the real world | <b>Q71 – Q80</b> |

Q1 – Q15

# Basic Full Stack Development

Foundations every full stack candidate must know

**BASIC FULL STACK DEVELOPMENT****Q1 – Q15****Q1****What is Full Stack Development?**

Full Stack development refers to front-end and back-end development of a website. This process includes designing the user interface of the website, server-side logic, database management, creation of APIs, and deployment of the application.

**Q2****What is a Full Stack Developer?**

A full stack developer is the developer who has the capability to develop both the client and server components of an application. A full stack developer will work with HTML, CSS, JavaScript, React, Java, Node.js, databases and APIs.

**Q3****What skills does a Full Stack Developer actually need?**

HTML, CSS, JS; a frontend framework; a backend language; databases; REST APIs; Git and testing basics. Technical skill only gets you so far, though being able to explain your thinking matters just as much.

**Q4****What is the difference between front end and back end development?**

The field of front end development involves user interfaces, while the domain of back end development includes server side scripting, databases, authentication, API calls, and more.

**Q5****What is HTML?**

The bones of a webpage. Every heading, image, form, and table starts out as an HTML element.

---

**BASIC FULL STACK DEVELOPMENT****Q1 – Q15**

---

**Q6****What is CSS?**

The layer that turns those bare bones into something people actually want to look at — color, spacing, fonts, and layout.

**Q7****What is JavaScript?**

JavaScript is the programming language used by developers to build web applications that make it possible to control user actions, form validations, web page contents, and interaction with the server.

**Q8****What is DOM?**

DOM stands for Document Object Model and is the object-oriented model of an HTML document. JavaScript uses the DOM model to access different elements on the web page.

**Q9****What are Semantic HTML elements?**

Semantic HTML elements are those that clearly define the purpose of their use. Examples include `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, and `<footer>` — elements that say exactly what they're for. They make pages easier to read for screen readers and search engines alike.

**Q10****What is Responsive Web Design?**

Responsive web design is what makes websites usable on computers, tablets, and mobile phones. This is achieved by utilizing fluid grids, media queries, and responsive images.

## BASIC FULL STACK DEVELOPMENT

## Q1 – Q15

Q11

**What is the difference between “==” and “===” in JavaScript?**

“==” compares after converting values, whereas “===” compares equality of value as well as type. For instance, `5 == "5"` returns true, while `5 === "5"` returns false.

Q12

**What is a variable in JavaScript?**

A variable refers to the entity that stores information to be used in the application. There are various types of variables such as `var`, `let`, and `const` in JavaScript. `let` and `const` are more common in modern programming.

Q13

**What is a JavaScript Function?**

A function is a piece of code that executes a particular task. It takes an input, performs some operations on the data, and gives an output.

Q14

**What is Asynchronous programming in JavaScript?**

Asynchronous programming is helpful in executing lengthy operations without interrupting other operations. Async programming is used in APIs, file processing, and database queries.

Q15

**What is a Promise in JavaScript?**

A Promise is an object that represents the result of an asynchronous task. Such an object can have different states: pending, fulfilled, or rejected. One can work with promises using `.then()`, `.catch()`, or `async/await`.

# 02

Q16 – Q30

## Front-End and React

UI frameworks, components, and browser fundamentals

## FRONT-END AND REACT

Q16 – Q30

Q16

**What is React?**

React is a JavaScript library used for building UI components. With the help of this library, one can use components to build single page applications.

Q17

**What is a React component?**

A React component is a part of the user interface that can be reused. Examples of React components include a navigation bar, product cards, forms, a dashboard, and buttons.

Q18

**What are props in React?**

Props are chunks of information passed from one component to another. They generally pass from a parent component to child components. Props are immutable.

Q19

**What is state in React?**

State is the data that can change during the lifecycle of the application. Examples of such data could be login status, the text of a search query, or chosen elements.

Q20

**What is the Virtual DOM?**

The Virtual DOM is an in-memory copy of the real DOM. React uses this concept to detect changes and update only the required elements of the user interface.

## FRONT-END AND REACT

Q16 – Q30

Q21

**What are React Hooks?**

Hooks are functions that allow functional components to use React capabilities such as state and side effects. Three common hooks are `useState`, `useEffect`, and `useContext`.

Q22

**What is `useEffect()`?**

`useEffect()` is a function that performs side effects in React components. Side effects performed using `useEffect()` include API calls, subscriptions, listeners, and cleanups.

Q23

**What is Angular?**

Angular is a framework that enables building web applications using the TypeScript programming language. Some features of Angular include components, routing, forms, dependency injection, and HTTP services.

Q24

**What is TypeScript?**

TypeScript is an advanced programming language that is a superset of JavaScript. It helps detect errors in advance and makes managing large projects easier.

Q25

**React vs Angular: what is the difference between them?**

React is primarily a UI library, while Angular is a full-fledged framework. React gives more flexibility in picking supporting tools, while Angular has all the features built into its framework.

## FRONT-END AND REACT

## Q16 – Q30

Q26

**What is event handling in JavaScript?**

Event handling allows the application to respond to user actions such as clicking and key pressing.

Q27

**What is event bubbling?**

When an event is fired on a child node, it bubbles up through the parent nodes. This helps with event handling through event delegation.

Q28

**What is localStorage?**

localStorage is a browser feature that stores key-value pairs. The stored data remains intact even when the browser is closed.

Q29

**What is sessionStorage?**

sessionStorage is also a key-value storage in the browser, but it stores data only for the current tab session.

Q30

**What is Lazy Loading?**

Lazy loading means loading application components only when they are required. It is used for applications that have many components or files.

# 03

Q31 – Q45

## Back-End and API

Servers, APIs, and how the pieces talk to each other

## BACK-END AND API

## Q31 – Q45

Q31

**What is Node.js?**

Node.js is a runtime environment for JavaScript that allows JavaScript to run outside the browser. It is usually used for building APIs, web servers, and applications.

Q32

**What is Express.js?**

Express.js is a minimalistic Node.js framework used for building web applications and RESTful APIs.

Q33

**What is middleware?**

Middleware is a piece of code used in the request-response cycle. It is responsible for tasks such as authentication, logging, validation, error handling, and request handling.

Q34

**What is an API?**

“API” stands for “Application Programming Interface,” a way for different software applications to communicate. For example, a frontend application may use one to access customer data.

Q35

**What is a REST API?**

A REST API is an approach to building web service architecture, using HTTP methods like GET, POST, PUT, PATCH, and DELETE.

## BACK-END AND API

Q31 – Q45

Q36

**What are HTTP methods?**

HTTP methods include GET (fetching data), POST (creating data), PUT (updating data), PATCH (partially updating data), and DELETE (deleting data).

Q37

**What is JSON?**

JSON stands for JavaScript Object Notation. It is a simple, compact format widely used for sending data between the front end and back end.

Q38

**What is authentication?**

Authentication verifies a user's identity. It includes methods such as password authentication, one-time passwords, authentication tokens, and OAuth.

Q39

**What is authorization?**

Authorization refers to the rights granted to an authenticated user to perform an action. For example, only an administrator may be authorized to remove other users.

Q40

**What is JWT?**

JWT stands for JSON Web Token, often used to transfer authentication details from a client to a server.

## BACK-END AND API

## Q31 – Q45

Q41

**What is CORS?**

CORS means Cross-Origin Resource Sharing. It governs whether a browser should allow a web application to make requests to other origins for resources.

Q42

**What are HTTP status codes?**

HTTP status codes are responses to a request. Common codes include 200 (success), 201 (created), 400 (bad request), 401 (unauthorized), 404 (not found), and 500 (server error).

Q43

**What is API pagination?**

API pagination splits a large data set into smaller pages of data. Rather than returning a thousand records at once, a limited number of records is returned per page.

Q44

**What is API rate limiting?**

Rate limiting sets a cap on the number of requests that can be made to an API within a given period of time.

Q45

**What is GraphQL?**

GraphQL is an API query language that lets the client request exactly the data it needs. Unlike REST APIs, clients can specify the exact shape of the response.

# 04

Q46 – Q60

## Database

Storing, relating, and querying data reliably

## DATABASE

Q46 – Q60

Q46

**What is a database?**

A database is a place where data is stored in a way that applications can create, read, update, and manage it.

Q47

**What is SQL?**

SQL stands for Structured Query Language, a programming language used to query, insert, update, and delete data in a database.

Q48

**What is a relational database?**

In a relational database, data is organized into tables of rows and columns, and these tables can be related to one another through keys.

Q49

**What is a primary key?**

A primary key is the column in a table that uniquely identifies each record – for example, Customer ID in a customers table.

Q50

**What is a foreign key?**

A foreign key is used to relate tables in a database. It usually corresponds to the primary key in another table.

## DATABASE

Q46 – Q60

Q51

**What is database normalization?**

Database normalization is the process of organizing data into related tables to reduce redundancy and improve data integrity.

Q52

**What does SQL JOIN mean?**

JOIN combines data from several tables using related columns. Common types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

Q53

**What is an index?**

An index is a technique that helps locate records in a database faster. However, indexing uses extra storage space and adds some cost when writing data.

Q54

**What is MongoDB?**

MongoDB is a NoSQL database. Unlike relational databases, MongoDB stores data in a flexible, JSON document-like form rather than tables.

Q55

**What is the difference between SQL and NoSQL?**

The key differences are in table structure and relationships. SQL databases use structured tables with defined relationships, while NoSQL databases are more flexible in structure.

## DATABASE

Q46 – Q60

Q56

**What is CRUD?**

CRUD stands for Create, Read, Update, and Delete — the four essential functions used to work with data in an application.

Q57

**What is a database transaction?**

A transaction is a set of database operations treated as a single logical operation, ensuring that either all related operations succeed or all fail.

Q58

**What is ACID?**

ACID stands for Atomicity, Consistency, Isolation, and Durability — properties that ensure reliable transactions in relational databases.

Q59

**What is ORM?**

ORM stands for Object-Relational Mapping. It lets developers manipulate database records using program objects instead of writing raw SQL statements.

Q60

**How can the performance of a database be improved?**

Database performance can be improved through indexing, query optimization, correct database design, pagination, and proper connection handling, among other techniques.

Q61 – Q70

## Java Full Stack Developer

OOP concepts and the Java enterprise stack

## JAVA FULL STACK DEVELOPER

## Q61 – Q70

Q61

**What is Java?**

Java is a programming language built on OOP concepts, widely used in enterprise-level applications, web development, APIs, and software systems.

Q62

**What are the OOP principles of Java?**

The four primary OOP concepts are encapsulation, inheritance, polymorphism, and abstraction. These concepts help build applications in a modular way.

Q63

**What is inheritance in Java?**

Inheritance lets a class borrow features and functions from another class. It helps achieve code reuse and models relationships between classes.

Q64

**What is polymorphism?**

Polymorphism refers to a scenario where the same interface or method shows different behavior depending on the object used. Overloading and overriding are examples of polymorphism in Java.

Q65

**What is encapsulation?**

Encapsulation bundles data and methods inside a class and controls access to that data from outside the class. Private attributes with getter/setter methods are a common way to achieve this in Java.

## JAVA FULL STACK DEVELOPER

## Q61 – Q70

Q66

**What is Spring Boot?**

Spring Boot is a free-to-use framework that simplifies Java application development with features like auto-configuration and an embedded web server.

Q67

**What is dependency injection in Spring?**

Dependency injection is a concept where one class receives another class rather than creating its own instance. This makes the application easier to test and manage.

Q68

**What is Hibernate?**

Hibernate is an Object-Relational Mapping (ORM) tool for Java that maps Java objects to relational database tables, reducing the amount of manual coding required.

Q69

**What is JPA?**

JPA stands for Java Persistence API — a specification for Java application persistence and object-relational mapping. Hibernate is one implementation of JPA.

Q70

**What is Maven?**

Maven is a tool that helps manage and automate the build process of a Java application, including managing libraries, compiling, testing, and packaging.

# 06

Q71 – Q80

## Git, DevOps, Security & Scenarios

Version control, deployment, and how you'd handle it in the real world

---

**GIT, DEVOPS, SECURITY & SCENARIOS****Q71 – Q80**

---

**Q71****What is Git?**

Git is a distributed version control system used to manage changes in code, work with branches, and collaborate with other programmers.

**Q72****What is GitHub?**

GitHub is an online platform that hosts Git repositories and supports collaboration on software projects, mostly used for pull requests, code review, and issue management.

**Q73****What is Docker?**

Docker is a container platform that makes applications consistent by packaging the application and all its dependencies together inside a container.

**Q74****What is CI/CD?**

CI/CD stands for Continuous Integration and Continuous Delivery/Deployment. It automates tasks such as testing and deploying applications.

**Q75****What is XSS?**

XSS, or cross-site scripting, is a security vulnerability where malicious scripts get injected into content viewed by other users. Data validation, output encoding, and good coding practices help mitigate this risk.

**GIT, DEVOPS, SECURITY & SCENARIOS****Q71 – Q80****Q76****What is CSRF?**

CSRF stands for Cross-Site Request Forgery. It tricks an authenticated user's browser into performing an action the user did not intend. CSRF tokens and secure cookies help protect against this attack.

**Q77****How would you go about implementing authentication?**

I would create authentication endpoints, validate inputs, hash passwords, verify credentials, and start a user session or token. Private endpoints should also handle authorization, alongside whatever the frontend does.

**Q78****How would you optimize a slow API?**

I would first identify the root cause from logs, database queries, network latency, calls to external services, or another part of the app — then optimize the database query, add caching or indexing, reduce response size, or optimize processing, depending on what the issue turns out to be.

**Q79****How would you handle an application with thousands of users?**

I would focus on efficient queries, caching, scaling, load balancing, asynchronous processing, optimized APIs, and careful use of resources — with the exact architecture shaped by traffic patterns and business requirements.

**Q80****How would you explain your project in a full stack interview?**

Start by introducing the problem your project solves. Then talk about the technologies, architecture, database, APIs, your specific tasks, and the technical problems you solved. Also be ready to describe a few key features in detail.



KNOW THESE ANSWERS?

## You could be shipping this code for a living.

Techpaathshala's Full Stack Development programme takes you from fundamentals to a portfolio of real, deployed projects — the same topics in this guide, but hands-on: HTML/CSS/JS, React, Node.js & Express, databases, Java & Spring Boot, Git, and deployment. Delivered by India's first work-based tech training institute.

EXPLORE THE PROGRAMME

**[www.techpaathshala.com](https://www.techpaathshala.com)**

[info@techpaathshala.com](mailto:info@techpaathshala.com)

Full Stack Development • Data Analytics • Data Science • GenAI / AI Engineering