- **Interpersonal Questions.**
    - Why we will hire you.
    - Tell me about your weak point.
    - Tell me about yourself.
    - what do you know about our company?
    - How confident are you that you can submit all tasks within the given deadline?
    - Where do you see yourself within 5 years?
    - What would you rate yourself in terms of coding?
    - Why did I choose this line of work, when I don't have a educational background in this line?
    - How did you get into web development?
    - Tell me about a recent project you worked on.
    - Tell me about the best project you worked on. Why is it best?
    - Have you worked in a team environment?
    - What is one weakness you are currently working on?
    - Why should we hire you? What makes you different from other candidates?
    - Why do you feel good about this role
    - What motivates you at work?

- **Technical Questions (HTML & CSS)**
    - Difference between position sticky and fixed.
    - what is css box model?
    - what is attribute?
    - what sematic html is and give some examples?
    - Explain semantic HTML and why it matters.
    - What is the box model?
    - Difference between absolute, relative, fixed, and sticky positioning.
    - Centering elements vertically and horizontally using CSS
    - Making responsive layouts for different screen sizes
    - Understanding the CSS box model (margin, border, padding, content)
    - Managing CSS specificity and overriding styles
    - Using semantic HTML tags correctly for structure and
    - What is the CSS box model
    - How can we responsive UI using row css
    - What are the Semantic Tags in html
    - How many type of css declaration
    - Difference between Flexbox and Grid? When should you use each?
    - What is the difference between inline, block, and inline-block elements?
    - Differentiate between the Inline and the Block elements in HTML.
    - What is special about Tailwind CSS?
    - Why would you use Tailwind instead of traditional CSS?
    - How do you customize Tailwind using the tailwind.config.js file?
    - What are utility classes, and why are they helpful?
    - How do you create responsive designs using Tailwind?
    - CSS কত প্রকার?

    - Seo ফ্রেন্ডলি করতে হলে একটা পেইজে কয়টি h tag ব্যাবহার করা হয়??
    - Difference between transform vs position for animations
- **Technical Questions (JS & TS)**
    - What is Closure?
    - What is promise in js.
    - Difference between var, let, and const.
    - What is scope?
    - What is hoisting?
    - what is es6?
    - Explain EventLoop.
    - What is hoisting?
    - Difference between Map and Foreach.
    - What is Hoisting in JavaScript?
    - What is the difference between let, var, and const?
    - Different between null and undefined?
    - What is DOM?
    - Difference between DOM and Virtual DOM?
    - Difference between var, let, and const.
    - What is hoisting?
    - What are Promises? Difference between .then() and async/await
    - What are array methods like map, filter, and reduce?
    - Why use TypeScript over JavaScript?
    - What are interfaces vs types?
    - What is any, unknown, and never?
    - Understanding asynchronous JavaScript (callbacks, promises, async/await)
    - Managing state and data flow in applications
    - Handling errors and debugging runtime issues
    - Understanding TypeScript types and interfaces
    - Converting JavaScript code to TypeScript and fixing type errors
    - Hoisting
    - this keyword
    - Is JS synchronous or not
    - Give me eaxmple of a loop
    - What is the DOM?
    - What is a callback function?
    - null vs undefined
    - What is the difference between call, apply & bind
    - What is hosting in JS
    - What is the difference between async & await
    - What is the promise
    - What is the pure function
    - What is JavaScript, and how is it different from Java?
    - What are arrow functions?
    - What is the event loop?
    - What is a higher-order function?
    - What is the difference between synchronous and asynchronous JavaScript?

    - Explain event bubbling and capturing?
    - What is a callback function?
    - What is optional chaining (?.)?
    - What are interface and type? What’s the difference?
    - What is TypeScript, and how is it different from JavaScript?
    - What is the spread operator?
    - What is the map method?
    - What is the filter method?
    - Why do we use the find method?
    - What is event bubbling?
    - What is event delegation?
    - What's the difference between let and const?
    - What is an arrow function?
    - What are callback functions?
    - Difference between == (double equal) and === (triple equal)?
    - What is an if/else statement and how does it work?
    - What are event bubbling and event delegation?
    - Difference between fetch and axios?
    - What is debouncing/throttling and when would you use them?
    - What are pure functions, and why are they useful?
    - What is the difference between synchronous and asynchronous code?
    - What is async await?
    - What is destructuring in JavaScript?
    - Explain the chaining method or What is the chaining method?
    - What is the NaN property in JavaScript?
    - Explain Undefined
- **Technical Questions (React & Next)**
    - What is React DOM and JS DOM?
    - Why we use next js
    - Explain useCallBack
    - Explain useState
    - Explain useMemo
    - What is virtual DOM?
    - Why do we use useMemo and useCallback?
    - How to manage cache in NextJS with the fetch api?
    - How to revalidate cache in fetch api?
    - Is NextJS store api result data in cache, and where is it stored?
    - Explain useState and useEffect.
    - What is Context API? When do you use it?
    - The difference between controlled and uncontrolled components.
    - What is memoization (React.memo, useMemo, useCallback)?
    - How do you handle forms in React?
    - What is a React hook?
    - What is Props? Why should developers use this, and why not?
    - What is Next.js? Why should I use this?
    - What is props dealing?

- What are components?
- Why do we use the effect?
- What is virtual DOM?
- How call the API.
- What is a hook?
- What is JWT?
- Explain some authentication methods.
- Why do we use useRef?
- What is Destructuring?
- What are client components?
- What are server components?
- What is the difference between SSR, SSG, and CSR in Next.js?
- What is the purpose of the app directory vs pages directory?
- How does API routing work in Next.js?
- What is dynamic routing in Next.js?
- How do you optimize performance in a Next.js application?
- What is prop drilling in React?
- What is the Context API?
- What is a component in React?
- What is useEffect?
- What is the virtual DOM?
- Why do we use JSX in React?
- What is the difference between HTML and React? Why do we use React instead of HTML?
- Explain the difference between controlled and uncontrolled components.
- What are React hooks? Explain useState and useMemo.
- How does React's reconciliation (virtual DOM) work?
- What are React keys, and why are they important?
- How do you handle state management for large apps?
- What causes unnecessary re-renders in React, and how do you prevent them?
- Difference between Prop and state?
- What are fragments in React?
- What is the difference between useEffect and useLayoutEffect?
- What is the useContext hook, and when should you use it?
- What is the useReducer hook, and when is it preferred over useState?
- What is the useRef hook?
- What are custom hooks in React, and how do you create one?
- When should you use Redux over Context API?
- What is a Higher-Order Component?
- Why do we use React?
- What is the Virtual DOM, and how does it work?
- What is a side effect in React, and how do we handle it?
- What is the difference between props and state?
- Explain the lifecycle methods of a React component.
- What is the difference between functional components and class components?
- What are controlled and uncontrolled components?
- What is the React Context API, and how is it used?

  - What is a React Component?
  - What is the difference between props and state?
  - SSR and CSR ki?
  - Main Feature of Next.js?
- **Technical Questions (Backend)**
  - What does CRUD stand for?
  - How do you find documents with a specific value?
  - How do you limit query results?
  - What is a projection in MongoDB?
  - Which library is used to connect MongoDB with Node.js?
  - How do you connect to MongoDB in Node.js?
  - What is Mongoose?
  - What is an index in MongoDB?
  - How do you create an index?
  - How do you create an index?
  - What is Node.js?
  - What is middleware in Node.js?
  - What is next in middleware?
  - What is global middleware?
  - Is Node.js single-threaded?
  - Why is Node.js suitable for I/O-heavy apps?
  - Can Node.js handle CPU-intensive tasks well?
  - What is Express.js?
  - How do you handle errors in Express?
  - How do you send a 404, 500, and 300 response?
  - What does next(err) do?
  - What is scaffolding in Express?
  - What is an API, and why do we connect it to a database?
  - What is MongoDB, and how is it different from SQL databases?
  - What is a collection and a document in MongoDB?
  - What is an ObjectId in MongoDB?
  - How do you perform CRUD operations in MongoDB?
  - Can you explain MongoDB indexing?
  - What is Express.js?
  - What are middlewares in Express?
  - What is Node.js, and why is it used for backend development?
  - What is the difference between synchronous and asynchronous code in Node.js?
  - How do you create a basic API endpoint in Express?