[Chapter 07 - Finding the Path](https://github.com/chetannada/Namaste-React/blob/main/Chapter%2007%20-%20Finding%20the%20Path)

**1] What are various ways to add images into our App?**

In React, images can be added in different ways depending on where they are stored:

1. **Importing images inside components (from src/)**
   * This is the most common method. The bundler (like Parcel, Vite, Webpack) processes the image, optimizes it, and includes it in the final build.

A computer code with text on it

AI-generated content may be incorrect.

2. **Using the public/ folder**

* Any file placed in the public folder is served as-is, without bundling. Useful for assets that don’t change.

A black background with white text

AI-generated content may be incorrect.

3. **Using external URLs**

* You can directly use image links hosted on a CDN or any server.

A computer screen with green text

AI-generated content may be incorrect.

👉 **In interview one-liner:**  
*“We can add images by importing them into components, placing them in the public folder, or using external URLs. The src/ method is preferred when bundling and optimization are needed, while public/ is used for static assets.”*

**2] What would happen if we do console.log(useState())?**

* The useState() hook in React always returns an **array of two elements**:
  1. The current state value.
  2. A function to update the state.

Example:

A screenshot of a computer program

AI-generated content may be incorrect.

👉 **Interview one-liner:**  
*“Calling useState() returns a pair: the current state and a setter function. That’s why destructuring [value, setValue] is the standard way to use it.”*

**3] How will useEffect() behave if we don’t add a dependency array?**

* If no dependency array is provided, the useEffect callback executes **after every render** (initial + updates).
* This is often not desired because it can cause **performance issues** or **infinite loops** if it updates state inside itself.

Example:

A computer screen with text

AI-generated content may be incorrect.

👉 **Interview one-liner:**  
*“Without a dependency array, useEffect runs after every render. With [], it runs only once (like componentDidMount).”*

**Extra Question -**

**🔹 What does "Mount" mean in React?**

* "Mount" means **placing a component into the DOM for the first time**.
* In simple words → when a component **first appears on the screen**, it is “mounted”.

**🔹 React Component Lifecycle (Class-based overview)**

React class components had 3 main phases:

1. **Mounting (Birth phase)** → when the component is created & inserted into DOM.
   * constructor()
   * componentDidMount() (runs once after first render).
2. **Updating (Growth phase)** → when props/state change, component re-renders.
   * componentDidUpdate()
3. **Unmounting (Death phase)** → when component is removed from DOM.
   * componentWillUnmount()

**🔹 Hooks equivalent (Functional Components)**

In functional components, **useEffect replaces lifecycle methods**.



✅ **Interview-ready one-liner:**  
*“Mount means the component is added to the DOM for the first time. In class components we used componentDidMount, but in functional components, the same behavior is handled by useEffect(() => {}, []).”*

**4] What is SPA?**

* **SPA (Single Page Application)** is a web application that loads a **single HTML file (index.html)**, and React dynamically updates the content without reloading the entire page.
* This makes navigation **faster** and gives a **native app-like experience**.

👉 **Interview one-liner:**  
*“An SPA is a web app that updates the UI dynamically within a single HTML file, avoiding full page reloads and improving user experience.”*

**5] Difference between Client-Side Routing vs Server-Side Routing**

SEO - (Search Engine Optimization)

| **Aspect** | **Client-Side Routing (CSR)** | **Server-Side Routing (SSR)** |
| --- | --- | --- |
| **Handled by** | Browser (React Router) | Server |
| **Page reloads?** | No reload, JS updates UI | Full page reload on navigation |
| **Speed** | Fast after first load | Slower, every request hits server |
| **SEO** | Needs extra handling (SSR/Next.js) | Naturally SEO-friendly |
| **Example** | React Router /about → shows About component | /about → server sends about.html |

👉 **Interview one-liner:**  
*“In Client-Side Routing, the browser dynamically updates components without reload using JavaScript. In Server-Side Routing, every route fetches a new HTML file from the server.”*