Introduction to React.js

Question 1: What is React.js? How is it different from other JavaScript frameworks and libraries ?

Ans.

What is React JS :

React.js is a JavaScript library developed by Facebook (Meta) for building fast, interactive, and component-based user interfaces—especially for single-page applications (SPAs). It follows a component-based architecture, where the UI is broken into reusable components. React uses a Virtual DOM, which allows efficient UI updates and high performance. **Key features of React.js :-**

- Component-based architecture
- Virtual DOM for faster rendering
- One-way (unidirectional) data flow
- JSX (JavaScript XML) for writing UI with HTML-like syntax

JSX (JavaScript XML) for writing UI with HTML-like syntax :

1. React vs Angular

Feature	React.js	Angular
Type	Library	Full framework
DOM	Virtual DOM	Real DOM
Data Flow	One way	Two way
Learning Curve	Easy	Hard
Language	JavaScript + JSX	TypeScript
Flexibility	Very flexible	Fixed Structure

2. **React vs Vue.js**

Feature	React.js	Vue.js
Type	Library	Framework
Syntax	JSX	HTML + JS
Learning Curve	Medium	Easy
Flexibility	High	Medium
Possibility	Very high	High

3. **React vs JQuery**

Feature	React.js	jQuery
Purpose	Build modern UI with components	DOM manipulation
DOM	Virtual DOM	Direct DOM manipulation
Architecture	Component-based	No architecture
Use Case	SPAs, large apps	Small DOM tasks

Why React is unique / different :-

React differs from traditional frameworks because:

1. Uses Virtual DOM → very fast updates

   Only changes the required parts of UI, not the whole page.

2. Component-based

   Reusable blocks → cleaner code, scalable apps.

3. Unidirectional data flow

   Data moves in one direction → predictable and stable.

4. Not a full framework

   You choose your own:

- routing (React Router)
- state management (Redux, Zustand, Recoil)
- styling (Tailwind, CSS-in-JS)

This makes React lightweight and highly flexible.

Question 2: Explain the core principles of React such as the virtual DOM and component based architecture.

Ans.

React is built on a few powerful principles that make it fast, scalable, and easy to develop user interfaces. The two most important concepts are:

1. **Virtual DOM (VDOM) :**

   What is DOM ?

   The DOM (Document Object Model) represents the web page structure. Updating the real DOM repeatedly is slow.

   What is Virtual DOM ?

   React creates a lightweight copy of the real DOM in memory → this is called the Virtual DOM.

   <u>How it works :-</u>

   1. When data changes, React updates the Virtual DOM, not the real DOM.
   2. React compares the Virtual DOM with the previous version → this process is called diffing.

Benefits :-

1. Faster performance (minimal real DOM updates)
2. Efficient rendering
3. Smooth UI updates
4. Better user experience

2. **Component-Based Architecture :**

React applications are built using components -- independent, reusable pieces of UI.

What is a component ?

A component is like a **small building block** of UI.

Example: Navbar, Button, Card, Sidebar, etc.

Types of components :

1. Functional Components
2. Class Components (older way)

Key features of components :

1. Reusable - write once, use many times
2. Independent - each component manages its own logic and UI
3. Organized - makes large apps easier to manage
4. Maintainable - small, isolated pieces are easier to debug

How components work:

1. Components receive props (data from parent)
2. Components manage state (internal data that changes)
3. React renders UI based on state + props

Benefits :

1. Reusability - saves time
2. Clean and modular code
3. Easy to scale for large applications
4. Better code organization

Question 3: What are the advantages of using React.js in web development ?
ANS.

Advantages of Using React.js in Web Development :

React.js is a popular JavaScript library for building fast, scalable, and interactive user interfaces. Its design focuses on performance, reusability, and maintainability.

Key Advantages of React.js :-

1. Component-Based Architecture
   - UI is broken into reusable components
   - Easy to manage, test, and maintain large applications

```
function Button() {
  return <button>Click Me</button>;
}
```

2. Virtual DOM for Better Performance
   - React uses a Virtual DOM to update only the parts of the UI that change.
   - Improves speed and efficiency compared to direct DOM manipulation.

3. Reusable Code
   - Components can be reused across the application
   - Reduces development time and code duplication

4. Fast Development & Easy Learning
   - Uses JSX, which is easy for HTML + JavaScript developers
   - Large community and rich documentation

5. One-Way Data Binding
   - Data flows in a single direction (parent → child)
   - Makes debugging easier and application behavior predictable

6. Strong Ecosystem & Community Support
   - Huge ecosystem of libraries (Redux, React Router, Axios, etc.)
   - Backed by Meta (Facebook) with long-term support

7. SEO-Friendly
   - Supports Server-Side Rendering (SSR) with frameworks like Next.js
   - Helps improve search engine visibility

8. Cross-Platform Development
   - Same concepts can be used for web (React.js) and mobile apps (React Native)
