**1. Define SPA and Its Benefits**

**SPA (Single-Page Application)** is a web application that loads a single HTML page and dynamically updates the content as the user interacts with the app, without refreshing the entire page.

**Benefits of SPA:**

* **Fast performance:** Only data is transferred, not entire pages.
* **Better user experience:** Smooth transitions and interactions without full reloads.
* **Reduced server load:** Less bandwidth used, as the same page is updated rather than reloaded.
* **Caching capabilities:** SPAs can store data locally to speed up future interactions.

**2. Define React and Identify Its Working**

**React** is an open-source JavaScript library developed by Facebook for building user interfaces, especially for SPAs. It focuses on creating reusable UI components.

**How React works:**

* Uses a **component-based architecture** to build UIs.
* Implements a **virtual DOM** to efficiently update only the parts of the page that change.
* Employs **unidirectional data flow**, meaning data flows from parent to child components.
* React updates the UI using a process called **reconciliation**, comparing old and new virtual DOMs.

3. Identify the Differences Between SPA and MPA

| **Feature** | **SPA (Single-Page Application)** | **MPA (Multi-Page Application)** |
| --- | --- | --- |
| **Page Load** | Loads a single HTML page | Loads a new HTML page for each interaction |
| **Speed** | Faster after initial load | Slower due to frequent reloads |
| **User Experience** | Smooth, like a desktop app | Less fluid due to full page reloads |
| **Navigation** | Handled by JavaScript routing (e.g., React Router) | Handled by the server |
| **SEO** | Harder to optimize without server-side rendering | Easier, since each page can be indexed |
| **Development** | Complex state management | Simpler, with server-side control |

**4. Explain Pros & Cons of Single-Page Application**

**Pros:**

* Faster performance after initial load
* Seamless user experience
* Reusable front-end code (components)
* Efficient use of network resources
* Good for building interactive web apps

**Cons:**

* SEO limitations (without server-side rendering)
* Initial load time can be longer
* Complex client-side routing and state management
* Browser history and analytics can be harder to manage

**5. Explain About React**

React is a **JavaScript library** for building user interfaces, especially for single-page applications. It allows developers to create **modular, reusable UI components**.

* Developed by **Facebook** and open-sourced in 2013.
* Enables developers to build **complex UIs** using declarative code.
* Promotes the use of **JSX** (JavaScript + HTML-like syntax).
* Often used with tools like Redux, React Router, and Next.js.

**6. Define Virtual DOM**

The **Virtual DOM (VDOM)** is a lightweight JavaScript representation of the actual DOM.

* React creates a virtual DOM in memory and uses it to track changes in the UI.
* When the state of an object changes, the virtual DOM updates, and then React compares it with the previous virtual DOM using a **diffing algorithm**.
* Only the **actual changes** are updated in the real DOM, improving performance.

**7. Explain Features of React**

**Key Features of React:**

1. **JSX (JavaScript XML):** A syntax extension that allows writing HTML-like code in JavaScript.
2. **Component-Based Architecture:** UI is broken into reusable components.
3. **Virtual DOM:** Efficient UI updates and rendering.
4. **Unidirectional Data Flow:** Ensures better control over data.
5. **Performance Optimization:** React efficiently updates and renders components.
6. **Declarative UI:** Makes code more predictable and easier to debug.
7. **React Hooks:** Enable functional components to use state and lifecycle features.
8. **Rich Ecosystem:** Supports routing, state management, and integration with other libraries.