**React Js HandsOn-1**

**1.Define SPA and its benefits**

A Single Page Application (SPA) is a web application or website that interacts with the user by dynamically rewriting the current page, rather than loading entire new pages from the server. In an SPA, after the initial page load, all necessary resources (HTML, CSS, JavaScript) are loaded once, and only data is fetched as needed using APIs (typically through AJAX or fetch).

**Benefits of SPA:**

1. **Faster Performance:**
   * Only data is sent and received after the initial load, not full pages. This makes the app feel faster and more responsive.
2. **Improved User Experience:**
   * No full page reloads; transitions are smoother, similar to desktop or mobile apps.
3. **Reduced Server Load:**
   * Server only sends data, not full HTML pages, reducing bandwidth usage and server work.
4. **Easier to Turn into a Mobile App:**
   * With frameworks like React Native or Ionic, SPAs can be repurposed as mobile apps easily.
5. **Better Caching:**
   * Since the app loads once, static resources (JavaScript, CSS) can be cached effectively, speeding up repeat visits.
6. **Efficient Development:**
   * Code can be modularized and reused more easily, especially when using frameworks like React, Angular, or Vue.

**2.Define React and identify its working**

React is an open-source JavaScript library used for building user interfaces (UIs), particularly single-page applications (SPAs). It was developed by Facebook and is widely used to create interactive, component-based UIs for web and mobile apps.

How React Works:

1. Component-Based Structure:
   * UI is divided into reusable pieces called components.
   * Each component is a JavaScript function (or class) that returns HTML-like code using JSX (JavaScript XML).
2. JSX (JavaScript XML):
   * A syntax extension that allows you to write HTML inside JavaScript.
   * Makes UI code more readable and maintainable.
3. Virtual DOM:
   * React creates a lightweight in-memory representation of the real DOM (called the virtual DOM).
   * When the state or props of a component change, React:
     1. Updates the virtual DOM.
     2. Compares it with the previous virtual DOM (using a diffing algorithm).
     3. Minimally updates the real DOM with only the necessary changes.
4. Unidirectional Data Flow:
   * Data flows in one direction (from parent to child components).
   * Components receive data via props and maintain internal data via state.
5. React Hooks (Functional Components):
   * Functions like useState, useEffect, etc., allow functional components to manage state and side effects.

**3. Identify the differences between SPA and MPA**

| Feature | SPA (Single Page Application) | MPA (Multi Page Application) |
| --- | --- | --- |
| Page Loading | Loads a single HTML page, updates content dynamically via JavaScript. | Loads a new HTML page from the server on each user action (like clicking a link). |
| User Experience | Smooth and fast experience with no full-page reloads. | Slower experience due to full page reloads. |
| Speed | Faster after the initial load since only data is exchanged. | Slower due to reloading of the entire page every time. |
| Routing | Handled on the client side (via JavaScript routers like React Router). | Handled on the server side, each URL maps to a different page. |
| Development Complexity | Easier for small to medium apps using frameworks like React, Vue, Angular. | Better suited for large, complex applications like e-commerce platforms. |
| SEO (Search Engine Optimization) | Harder, requires extra effort like server-side rendering (SSR) or pre-rendering. | Better SEO support by default, as each page has its own URL and content. |
| Initial Load Time | Slower initially because the full app is loaded at once. | Faster initial load, since only the needed page is loaded. |
| Examples | Gmail, Facebook, Twitter (modern versions) | Amazon, Wikipedia, Government portals |

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

| **Aspect** | **Pros** | **Cons** |
| --- | --- | --- |
| **Performance** | Fast interactions after initial load | Slow initial loading due to large JavaScript bundle |
| **User Experience** | Smooth, app-like experience without full page reloads | Back/forward navigation requires extra handling |
| **Development** | Reusable components, easier maintenance with frameworks like React/Vue | Complex state management in large apps |
| **SEO (Search Engines)** | Can be optimized with SSR or pre-rendering | Poor SEO by default |
| **Server Load** | Less load on server after initial request | More client-side processing required |
| **Offline Capability** | Good offline support with caching and service workers | Requires advanced setup for offline support |
| **Security** | Better control over client-side logic | Prone to XSS attacks if not handled securely |
|  |  |  |

1. **Explain about React**

ReactJS is a component-based JavaScript library used to build dynamic and interactive user interfaces. It simplifies the creation of single-page applications (SPAs) with a focus on performance and maintainability.

* It is developed and maintained by Facebook.
* The latest version of React is React 19.
* Uses a virtual DOM for faster updates.
* Supports a declarative approach to designing UI components.
* Ensures better application control with one-way data binding.

**6 .Define Virtual DOM**

The Virtual DOM (VDOM) is a lightweight, in-memory representation of the real DOM (Document Object Model). It helps React manage UI updates more efficiently by keeping a virtual version of the UI in memory. When changes occur, React updates only the necessary parts of the real DOM, instead of re-rendering everything**.**

1. **Explain features of React**

Component-Based

JSX (JavaScript XML)

Virtual DOM

Unidirectional Data Flow

React Hooks

High Performance

Declarative UI

Large Ecosystem

Cross-Platform Support

SEO Friendly (with SSR)