**REACT – ASSIGNMENT**

**1. Define SPA and its Benefits**

**SPA (Single-Page Application)** 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. This provides a more seamless and responsive user experience.

**Benefits:**

* **Faster load times** after the initial page load.
* **Smooth user experience** similar to a desktop app.
* **Reduced server load** as fewer full-page requests are made.
* **Efficient front-end routing** using JavaScript frameworks like React, Angular, or Vue.

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

**React** is an open-source JavaScript library developed by Facebook for building user interfaces, particularly single-page applications where data changes over time.

**Working:**

* **Component-Based:** UI is broken down into reusable components.
* **Virtual DOM:** React creates a virtual representation of the UI in memory, and syncs it with the real DOM using a process called **reconciliation**.
* **Unidirectional Data Flow:** Data flows from parent to child components using props.

**SPA vs MPA – Key Differences in Points**

1. **Page Reloading**
   * **SPA:** Loads a single HTML page and updates dynamically without full reloads.
   * **MPA:** Loads a new HTML page from the server for each interaction.
2. **Speed**
   * **SPA:** Faster after the initial load due to partial page updates.
   * **MPA:** Slower as each user interaction triggers a full page reload.
3. **User Experience**
   * **SPA:** Smooth and app-like experience.
   * **MPA:** Traditional website experience with visible page reloads.
4. **Server Load**
   * **SPA:** Reduces server load by fetching only required data via APIs.
   * **MPA:** Increases server load due to full-page reloads on every request.
5. **SEO (Search Engine Optimization)**
   * **SPA:** More challenging without server-side rendering or special tools.
   * **MPA:** SEO-friendly as each page has its own URL and metadata.
6. **Routing**
   * **SPA:** Handled on the client side using JavaScript (e.g., React Router).
   * **MPA:** Routing is handled by the server.
7. **Development Complexity**
   * **SPA:** More complex to build initially, especially with dynamic routing and state management.
   * **MPA:** Easier to develop and maintain with basic HTML/CSS/JS or server-side rendering.
8. **Initial Load Time**
   * **SPA:** Takes longer initially as the entire app is loaded at once.
   * **MPA:** Faster initial load as only one page is fetched.
9. **Data Handling**
   * **SPA:** Uses AJAX or Fetch API to communicate with the server asynchronously.

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

**Pros:**

* Fast and responsive UX.
* Reusable code with components.
* Less bandwidth usage after first load.

**Cons:**

* Poor SEO without additional setup.
* Initial load can be heavy.
* Requires JavaScript to function properly.

**5. Explain About React**

React is a JavaScript library for building fast, scalable, and interactive user interfaces. It allows developers to build complex UIs from small and isolated pieces of code called components. React is commonly used in modern web development, especially for SPAs.

**Key Concepts:**

* JSX (JavaScript XML)
* Components (Functional & Class-based)
* State and Props
* Lifecycle Methods (for class components)
* Hooks (for functional components)

**6. Define Virtual DOM**

The **Virtual DOM (Document Object Model)** is a lightweight JavaScript object which is a copy of the real DOM. React uses it to efficiently update and render components.

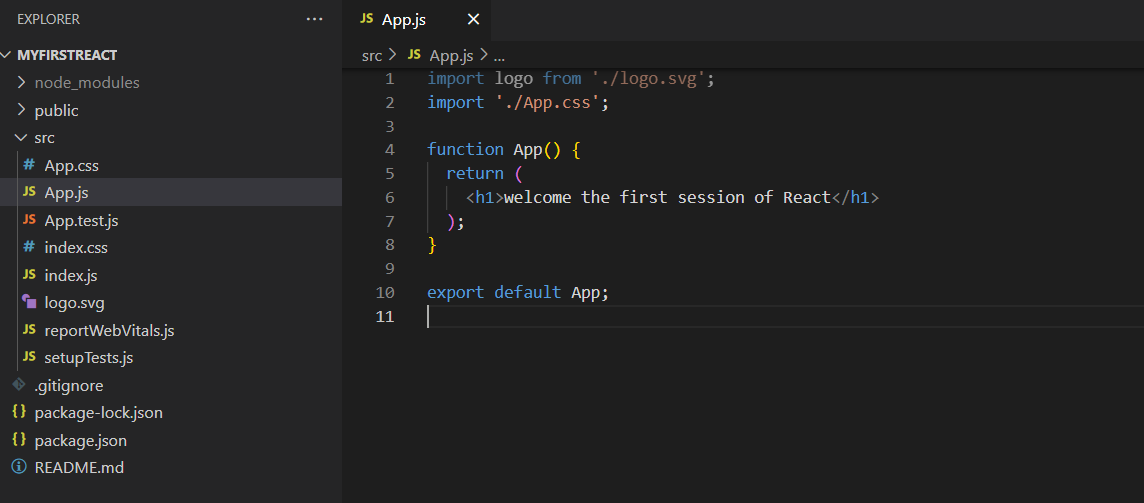
**How it works:**

* React updates the virtual DOM whenever state or props change.
* It compares the new virtual DOM with the previous one (diffing).
* React updates only the parts of the real DOM that changed (reconciliation), making it faster than traditional DOM manipulation.

**7. Explain Features of React**

* **JSX Syntax:** HTML-like syntax in JavaScript.
* **Component-Based Architecture:** Encourages reuse and better organization.
* **Unidirectional Data Flow:** Ensures data consistency.
* **Virtual DOM:** Optimizes rendering.
* **Hooks:** Functional component features like useState, useEffect.
* **React Native:** Enables cross-platform mobile development.
* **Strong Community & Ecosystem:** Vast library support, tools, and developer help.

**HANDS-ON: SET UP A REACT ENVIRONMENT & USE CREATE-REACT-APP**



**OUTPUT:**

