**Objectives**

This lab aims to help learners:

* Define SPA (Single Page Application) and its benefits
* Define React and identify how it works
* Identify the differences between SPA and MPA (Multi Page Application)
* Explain pros and cons of SPA
* Define Virtual DOM
* Explain key features of React
* Learn to set up the React environment and use Create React App (CRA)

**What is SPA?**

A Single Page Application (SPA) is a web application that loads a single HTML page and dynamically updates content without refreshing the entire page.

**Benefits of SPA:**

* Faster navigation (no full page reloads)
* Better user experience
* Reduces server load
* Responsive and interactive feel

**What is React?**

React is a JavaScript library developed by Facebook for building fast and interactive user interfaces (especially SPAs).

**How React Works:**

* Uses a component-based architecture.
* Each component represents a part of the UI.
* React updates the Virtual DOM (a lightweight copy of the real DOM) and then updates only the changed parts in the actual DOM.

**SPA vs MPA**

A Single Page Application (SPA) is a type of web application that loads a single HTML page and dynamically updates content without reloading the entire page. When a user interacts with the app (clicking a link, submitting a form), only the required part of the page updates using JavaScript (usually via libraries like React, Vue, Angular, etc.).

A Multi Page Application (MPA) consists of multiple HTML pages. Every user interaction (like clicking a link) sends a new request to the server, and the server responds by loading a new HTML page entirely. Each page is rendered independently and may require a full refresh of the browser.

**Key Differences:**

1. **Page Reloads:**
   * In SPAs, there is no full page reload; only parts of the content are updated.
   * In MPAs, every interaction like clicking a link results in a full page reload.
2. **Speed and Performance:**
   * SPAs provide faster navigation because resources are loaded once at the beginning.
   * MPAs are slower as each page change needs a server round trip and full reload.
3. **User Experience:**

* SPAs offer a more seamless and app-like experience, smoother transitions.
* MPAs might feel disconnected due to frequent page reloads.

1. **Complexity:**

* SPAs are generally more complex to build and manage, especially with routing and state management.
* MPAs are simpler to build but can become bulky with repetitive code for common layouts.

**Pros of SPA (Single Page Application)**

1. **Fast User Experience:**  
   One of the biggest advantages of SPAs is their speed. Since the application loads only once and then dynamically updates the content without refreshing the whole page, it provides a faster and smoother experience to the user. Only the necessary data is fetched, which reduces the load time significantly after the initial load.
2. **Improved Performance After First Load:**  
   After the initial HTML/CSS/JavaScript assets are loaded, SPA apps rely heavily on APIs and JavaScript rendering. This makes the subsequent navigation almost instant, enhancing user satisfaction.
3. **Better User Experience:**  
   SPAs are more interactive and fluid, making the user feel like they are using a desktop or mobile app. The transitions between views are seamless without page reloads, which helps in building modern and intuitive UIs.
4. **Efficient Development:**  
   With SPAs, front-end and back-end development can be separated easily. Developers can build the UI using frameworks like React, Angular, or Vue, while back-end developers focus solely on APIs. This improves modularity and maintainability of the code.
5. **Reusable Components:**  
   Modern SPA frameworks promote the use of reusable components, which makes it easier to manage UI logic and avoid code duplication.

**Cons of SPA**

1. **SEO Challenges:**  
   Since SPAs load content dynamically through JavaScript, search engine crawlers (like Google) may struggle to index the content unless additional techniques like server-side rendering (SSR) or pre-rendering are used. This makes SPAs less SEO-friendly out of the box.
2. **Initial Load Time:**  
   Although SPAs are fast after the initial load, the first time you open the application, it may take longer to load, because all JavaScript, CSS, and assets are downloaded at once. This can impact performance on slower networks or low-end devices.
3. **Browser History and Navigation Issues:**  
   Handling back and forward navigation in SPAs can be tricky. Developers must implement custom logic using libraries like React Router, otherwise the default browser behavior won’t work as expected.
4. **Security Concerns:**  
   SPAs are more exposed to cross-site scripting (XSS) attacks since a lot of logic runs on the client side. It requires careful handling of input sanitization and validation.
5. **JavaScript Dependency:**  
   SPAs rely heavily on JavaScript. If a user has disabled JavaScript in their browser, the application will not function at all. Also, if there are bugs in JavaScript code, the whole app may crash or behave unpredictably.

**What is Virtual DOM?**

The Virtual DOM (VDOM) is a lightweight, in-memory representation of the real DOM (Document Object Model). It is a core concept in frameworks like React, where it is used to optimize rendering performance and make updates more efficient.

Instead of directly interacting with the real DOM—which can be slow and expensive—a virtual copy of the DOM is maintained in memory. Whenever the state of a component changes, a new virtual DOM tree is created and compared with the previous one to determine what has changed.

**Core Features of React**

React is a JavaScript library for building user interfaces, especially for single-page applications (SPAs). It helps developers create reusable UI components and handle dynamic data efficiently.

**1. JSX (JavaScript XML)**

JSX is a syntax extension for JavaScript that lets you write HTML-like code inside JavaScript. It makes your code more readable and allows you to describe what the UI should look like.

**2. Components**

React is component-based, meaning the UI is built from independent, reusable pieces called components. Components can be:

* Functional Components (using hooks)
* Class Components (older, use lifecycle methods)

**3. Virtual DOM**

React uses a Virtual DOM to improve performance. When data changes, React:

* Creates a virtual DOM
* Compares it to the previous one (diffing)
* Updates only the changed parts in the real DOM (reconciliation)

This results in faster UI rendering.

**4. One-Way Data Binding**

React follows unidirectional (one-way) data flow. Data flows from the parent component to the child using props, making data management predictable.

**5. State Management**

React components can have internal state using the useState hook (in functional components). State represents dynamic data that affects what’s shown on the UI.

**7. React Router (SPA Navigation)**

React Router is used for client-side routing. It allows navigation between different pages/components without reloading the page.

**React Application**



