React and Single-Page Application (SPA) – Hands-on Lab

# Defined Objectives

## 1. Define SPA and its benefits

SPA (Single-Page Application): A web application that loads a single HTML page and dynamically updates the page as the user interacts with the app.  
Benefits:  
- Faster user interactions (no full-page reload)  
- Better performance after initial load  
- Smooth and app-like user experience  
- Reduces server load

## 2. Define React and identify its working

React: A JavaScript library created by Facebook for building user interfaces.  
How it works:  
- Uses a component-based architecture  
- Renders components efficiently using a Virtual DOM  
- Updates the UI in response to data changes using a reactive approach

## 3. Identify the differences between SPA and MPA

|  |  |  |
| --- | --- | --- |
| Feature | SPA | MPA (Multi-Page Application) |
| Page Reload | No (uses dynamic routing) | Yes (each interaction triggers reload) |
| Speed | Fast after initial load | Slower due to repeated server requests |
| Navigation | Client-side | Server-side |
| SEO Support | Limited (without SSR) | Strong SEO support |
| Complexity | Needs advanced JS setup | Traditional server setup |

## 4. Explain Pros & Cons of SPA

Pros:  
- Fast and responsive UI  
- Reduced bandwidth usage  
- Great for mobile apps  
  
Cons:  
- Poor SEO without SSR  
- Initial load can be large  
- JavaScript dependency (won’t work well with JS disabled)

## 5. Explain about React

React is a lightweight JavaScript library focused on building rich, dynamic, and responsive user interfaces.  
It allows developers to create reusable UI components and manage the view layer efficiently.

## 6. Define Virtual DOM

The Virtual DOM is a lightweight JavaScript representation of the actual DOM.  
React uses it to detect and apply only the minimal required changes to the real DOM, improving performance significantly.

## 7. Explain Features of React

- JSX (JavaScript XML): Allows HTML inside JavaScript  
- Virtual DOM: Efficient rendering  
- Components: Reusable and independent UI blocks  
- Hooks: Simplified state and side-effect management  
- One-Way Data Binding: Ensures better control over data flow  
- React Router: Enables client-side routing in SPAs

# Hands-on Lab Instructions

In this hands-on lab, you will learn how to:

· Set up a react environment

· Use create-react-app

# Prerequisites

The following is required to complete this hands-on lab:

· Node.js

· NPM

· Visual Studio Code

# Notes

Estimated time to complete this lab: 30 minutes.

# Steps to Create Your First React App

1. To create a new React app, install Node.js and NPM from the following link: https://nodejs.org/en/download/

2. Install Create-react-app by running the following command:  
  
npm install -g create-react-app

3. To create a React Application with the name of “myfirstreact”, type the following command:  
  
npx create-react-app myfirstreact

4. Once the App is created, navigate into the folder of myfirstreact by typing the following command:  
  
cd myfirstreact

5. Open the folder of myfirstreact in Visual Studio Code  
  
code .

6. Open the App.js file in Src Folder of myfirstreact

7. Remove the current content of “App.js”

8. Replace it with the following:  
  
import React from 'react';  
  
function App() {  
 return (  
 <div>  
 <h1>Welcome to the first session of React</h1>  
 </div>  
 );  
}  
  
export default App;

9. Run the following command to execute the React application:  
  
npm start

10. Open a new browser window and type “localhost:3000” in the address bar

# Screenshot of Output

