**1. Define SPA and Its Benefits**

**SPA (Single Page Application):**  
A web application that loads a single HTML page and dynamically updates content as the user interacts, without refreshing the page.

**Benefits of SPA:**

* Faster load time after initial load
* Smooth user experience
* Less bandwidth usage
* Easier to debug with Chrome/Firefox dev tools
* Reusable front-end components

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

**React:**  
A JavaScript library for building user interfaces, maintained by Facebook. It allows developers to create reusable UI components.

**How it works:**

* Uses a **virtual DOM** for efficient UI updates
* Components are the building blocks
* React re-renders only the components that change

**3. Differences Between SPA and MPA**

| **Feature** | **SPA** | **MPA (Multi Page Application)** |
| --- | --- | --- |
| Page Reload | No reload | Full reload on each navigation |
| Speed (after load) | Fast | Slower |
| Backend Dependency | Mostly front-end heavy | Backend handles routing |
| SEO | More challenging (needs SSR) | Easier for search engines |
| User Experience | Smoother | Less smooth |

**4. Pros & Cons of SPA**

**Pros:**

* Fast navigation
* Better user experience
* Code reusability (components)
* Easy debugging

**Cons:**

* SEO limitations (unless SSR is used)
* Slower initial load
* Heavier use of JavaScript
* Can be complex to secure

**5. Explain About React**

React is a **component-based library** used to create **interactive UIs**. It manages the **view layer** of web apps and allows creating large web applications that can change data without reloading the page.

**6. Define Virtual DOM**

**Virtual DOM (VDOM):**  
An in-memory representation of the real DOM. React compares the new VDOM with the previous version using **diffing algorithms**, and updates only the changed parts in the real DOM. This enhances performance.

**7. Features of React**

* Component-based architecture
* Virtual DOM
* One-way data binding
* JSX (JavaScript + HTML)
* Declarative UI
* Reusable components
* Hooks for state and lifecycle

**Lab Steps**

**1. Install Create-React-App**

bash

CopyEdit

npm install -g create-react-app

**2. Create New React App**

bash

CopyEdit

npx create-react-app myfirstreact

**3. Navigate into the App Folder**

bash

CopyEdit

cd myfirstreact

**4. Open in Visual Studio Code**

bash

CopyEdit

code .

**5. Edit App.js**

* Navigate to src/App.js
* Replace the contents with:

jsx

CopyEdit

import React from 'react';

function App() {

return (

<div>

<h1>Welcome to the first session of React</h1>

</div>

);

}

export default App;

**6. Run the App**

bash

CopyEdit

npm start

**7. View in Browser**

* Open your browser and go to:  
  [**http://localhost:3000**](http://localhost:3000)

You should see:

**Welcome to the first session of React**

