Objectives

- **Define SPA and its benefits**

  SPA is a web application that loads a single HTML page and dynamically updates the content using javascript without reloading the entire page.

  Benefits:

  1. Faster page loading
  2. Smooth user experience
  3. less server load
  4. Efficient client-side routing(no full-page reload)

- **Define React and identify its working**

  React is a JavaScript library developed by Facebook to build fast and interactive user interfaces (UI), especially for SPAs.

  React Working Principle:

  Uses components (reusable UI blocks)

  Uses JSX (JavaScript + HTML syntax)

  Implements Virtual DOM for fast UI updates

- **Identify the differences between SPA and MPA**

Feature	**SPA**	**MPA**
Page Reload	No	Yes
Speed	Faster after first load	Slower due to full reload
Routing	Client-side	Server-side
Development Effort	Easier for dynamic apps	Better for large content sites

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

Pros:

- Fast and responsive
- Great user experience
- Easy API integration
- Reduces server load

Cons:

- SEO challenges (can be solved using SSR)

- Longer initial load time
- Browser history & analytics handling may need extra work

- **Explain about React**

  React is a **JavaScript library** for building user interfaces, especially for **single-page applications** where you want smooth interactivity and dynamic content without refreshing the page.

  - Developed by **Facebook** in 2013
  - It's focused on the **view layer** of web applications (think of it as the front-end look and feel)
  - Based on the concept of **components** — reusable building blocks like LEGO bricks

  Key Concepts

  - **Components**
    Encapsulated UI pieces that manage their own logic and rendering. Examples: buttons, forms, headers.
  - **JSX (JavaScript XML)**
    Looks like HTML but lives inside JavaScript—makes writing UI code easier and more intuitive.
  - **Props (Properties)**
    Data passed from parent components to child ones. Like sending ingredients to a recipe!
  - **State**
    Internal data that a component can change over time (like tracking a counter or form input).
  - **Virtual DOM**
    React creates a virtual representation of the actual DOM and updates it efficiently—no full page reloads!

  Why Use React?

  - Lightning-fast user experience
  - Clean code and reusable components
  - Huge community and rich ecosystem
  - Supported by tools like React Router, Redux, and frameworks like Next.js

  Example

  ```
  function Greeting({ name })
  {
    return <h1>Hello, {name}!</h1>;
  }
  ```

  This simple React component greets the user by name.

- **Define virtual DOM**

  The **Virtual DOM (VDOM)** is a *lightweight copy* of the actual **DOM (Document Object Model)** used in web development.

  - The DOM represents the structure of your web page as objects — like a blueprint.
  - React doesn't directly change the real DOM with every user interaction. Instead, it updates the virtual DOM first

- **Explain Features of React**

1. **Component-Based Architecture**
   - Breaks UI into reusable pieces called *components*
   - Promotes cleaner code and separation of concerns
   - Think of it like creating individual puzzle pieces 🧩 that come together to form a complete picture
2. **JSX (JavaScript XML)**
   - A syntax extension that lets you write HTML-like code within JavaScript
   - Makes UI code easy to understand and write

     const element = <h1>Hello, world!</h1>;
3. **Virtual DOM**
   - Creates a virtual representation of the UI
   - Efficiently updates only parts of the DOM that change
   - Improves performance and responsiveness
4. **Unidirectional Data Flow**
   - Data flows one way—from parent to child components through **props**
   - Easier to debug and track changes in your app
5. **Hooks (like useState, useEffect)**
   - Introduced in React 16.8 for managing state and side-effects in functional components
   - Simplifies component logic and enhances readability
6. **React Native Support**
   - Lets you build mobile apps using React
   - Code once in React, deploy to iOS and Android 📱
7. **Rich Ecosystem**
   - Integrates easily with tools like Redux, React Router, and Next.js
   - Massive community and tons of libraries, tutorials, and support
8. **Server-Side Rendering (SSR)**
   - Improves performance and SEO by rendering React components on the server (e.g., with Next.js)

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

- Set up a react environment
- Use create-react-app

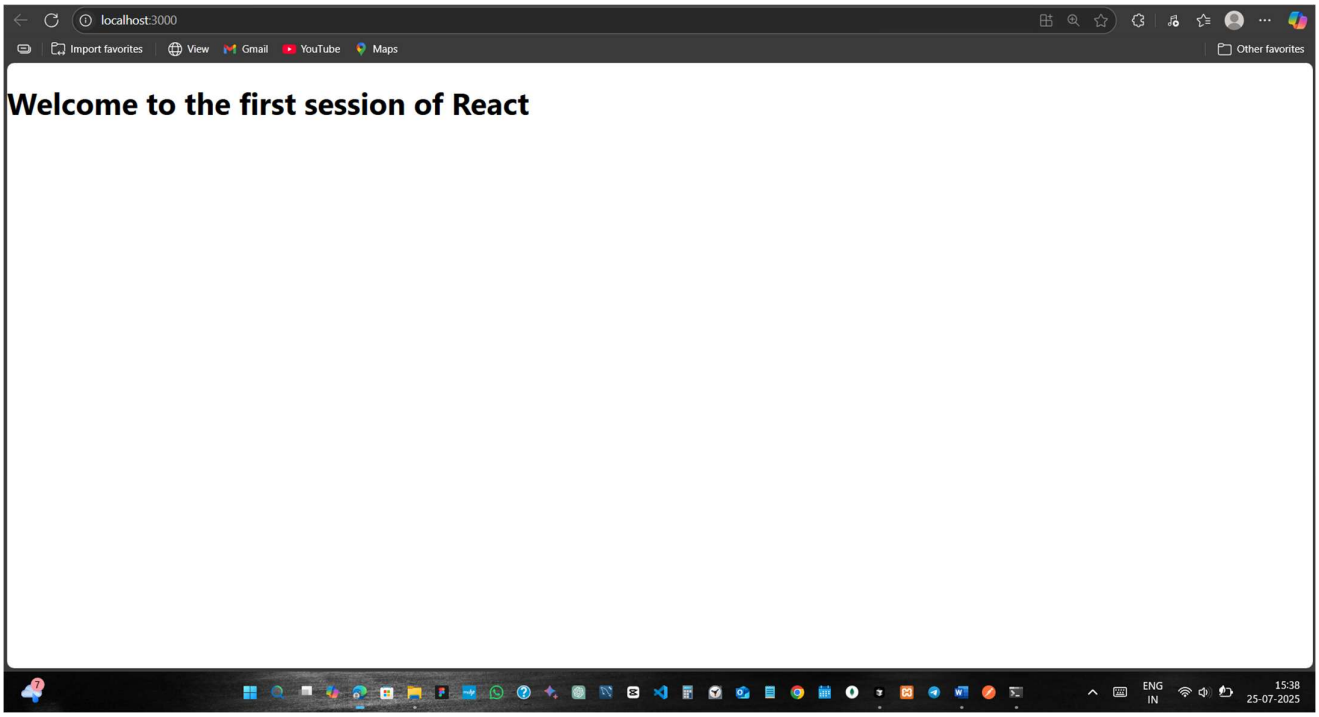
Prerequisites

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

- Node.js
- NPM
- Visual Studio Code

Result:

http://localhost:3000



Notes

Estimated time to complete this lab: **30 minutes.**

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

1. To create a new React app, Install Nodejs and Npm from the following link:

https://nodejs.org/en/download/

2. Install Create-react-app by running the following command in the command prompt:

```
C:>npm install -g create-react-app
```

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

```
C:>npx create-react-app myfirstreact
```

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

```
C:>cd myfirstreact
```

5. Open the folder of myfirstreact in Visual Studio 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:

```
function App() {
  return (
    <h1> Welcome the first session of React </h1>
  );
}
```

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

```
C:\myfirstreact>npm start
```

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

Welcome the first session of React