WEEK 6

Lab Activity: Introduction to React (States, Hooks)

Objective: The objective of this lab is to introduce students to the fundamentals of React, including state management and the use of React Hooks.

Materials Needed:

1. Computer with Internet access
2. A text editor or Integrated Development Environment (IDE)
3. Node.js and npm (Node Package Manager) installed

Duration: Self-paced

Instructions:

Part 1: Introduction to React

1. **What is React?:**

- ☐ Start by researching what React is and why it's essential in modern web development. You can find introductory articles and videos online.

2. **Setting Up Your Environment:**

- ☐ Ensure that you have Node.js and npm (Node Package Manager) installed on your computer. You can download them from https://nodejs.org/.

Part 2: Creating a React Project

1. **Creating a React App:**

- ☐ Open your terminal or command prompt.
- ☐ Use the following command to create a new React application. Replace my-react-app with your preferred project name.

```
npx create-react-app my-react-app
```

2. **Navigating to the Project Directory:**

- ☐ Move into your project directory using the cd command.

```
cd my-react-app
```

Part 3: Understanding React States

1. **React Component Structure:**

- ☐ Explore the structure of a React component, including the import statement, component definition, and render method.

2. **State in React:**

- □ Learn about state in React and how it enables components to manage and store data.
- □ Modify the default state of a component to include custom data.

3. **Rendering State Data:**

- □ Display the state data within your component's render method.

Part 4: Introduction to React Hooks

Introduction to Hooks:

- □ Understand the concept of React Hooks and how they simplify state management and side effects.
- □ Learn about common React Hooks like **useState()**.

Using useState():

- □ Implement the **useState()** Hook in your React component to manage and update state.

Part 5: Creating a Simple Interactive Component

1. **Create an Interactive Component:**

- □ Develop a simple React component that includes a button and a counter.
- □ Use React state to manage and display the count value.

2. **Adding Functionality:**

- □ Implement the functionality to increment the count when the button is clicked.

Part 6: Recap and Assignment

1. **Recap and Assignment:**

- □ Summarize what you've learned about React, states, and React Hooks in this lab.
- □ As an assignment, enhance your interactive React component by adding a decrement button and implementing its functionality.

Conclusion:

This self-paced lab activity provides you with an introductory understanding of React, including state management and the use of React Hooks. You'll set up a React project, create a simple interactive component, and gain hands-on experience with React's core concepts. Completing this activity will prepare you for more advanced React development in the course.