Lab 5

Section A: Mobile Web Apps with React

The objective of this lab is to help you understand the core concepts of React, a popular component-based web framework by analyzing and modifying a simple counter application.

Task 1: Code Analysis

Examine the provided code for the React Counter App. Identify the following key elements and be prepared to discuss them with your instructor.

1. What is the name of the main component of the application?
2. What does the `useState` hook do? What are the two variables it returns?
3. How is a button's `onClick` event handled in the code?

Task 2: Code Modification

Now, you will modify the application. Follow these steps to complete the task.

1. **Change the Heading:** Find the `<h1>` tag in the code and change the text inside to something new, for example, "Welcome to My First React App!".
2. **Change the Button Color:** Find the 'Increment' button. Change the class `bg-blue-600` to a new Tailwind CSS color, such as `bg-purple-600`.
3. **Use the Reset Button:** The code already includes a 'Reset' button that calls a `handleReset` function to set the counter back to 0. Click the button to see how it works.

Task 3: Lab Reflection

After completing the tasks, reflect on these questions to solidify your understanding.

1. How does a React component differ from a standard HTML file? What is the main benefit of using a component-based approach?
2. What is the role of `useState` in this application? How does it make the counter interactive?
3. Why is using a framework like React better for building complex mobile web applications than just using plain HTML and JavaScript?

Task 4: Creative Extension

Now for the creative part! Choose **one** of the following tasks to add a personal touch to your app.

1. **Personalize the App:** Change the entire color scheme to your favorite colors. You can modify the background color, the counter color, and the button colors. Use different Tailwind classes like `bg-teal-500`, `text-yellow-400`, etc., to create a unique look and feel.
2. **Add a New Feature:** Add a new button that **decrements** the counter. The button should have its own unique style and function, ensuring the counter doesn't go below zero.

Section B – Flight Booking

Objective

To build a simplified, single-page flight booking application using the React framework. This lab will apply concepts of component-based design, state management, and event handling in a more practical context.

Task 1: Conceptualization & Component Design (30 minutes)

Before writing any code, plan the structure of the application. Think about the different sections of the app as individual, reusable components.

Proposed Components:

- App (Main container component)
- BookingForm (Handles destination and seat selection)
- BookingConfirmation (Displays the final booking details and a confirmation message)

Task 2: Building the Core Application (60 minutes)

Analyse the flight.html file and answer the following questions to reflect on the experience.

1. How did the use of useState make the application dynamic? What would have been required to achieve the same functionality with plain JavaScript and HTML?
2. What is the purpose of the useEffect hook in this application? Why is it useful for calculating the total price?
3. Why is the generatePDF function a good example of how React can be extended with external libraries to add new functionality? What are some other features you could add to this app using different libraries?

Task :Follow these steps to customize the user interface.

1. **Change the Color Scheme:**
   - Find the main container `<div>` with the `bg-white dark:bg-gray-800` classes and change them to a different color combination you like. For example, `bg-teal-100 dark:bg-teal-900`.
   - Change the background color of the `<body>` to match your new theme.
   - Change the text color of the buttons and headings to complement your new theme.
2. **Add a Visual Element:**
   - Add an emoji or a small icon to the main heading. You can use an emoji like a plane ✈️ or a globe 🌍. You can refer to the Emoji Site.
   - Consider adding a small icon or emoji to the button labels.
3. **Enhance the Buttons:**
   - Find the "Confirm Booking" button and add a hover effect. Tailwind CSS classes like `hover:shadow-xl` and `hover:scale-110` can make the button pop when the user interacts with it.
   - Apply a similar hover effect to the "Download PDF Confirmation" button.

Section C: Attempt the Test Your Knowledge Quiz in Moodle with at least 80% accuracy.

END OF LAB 5