Name: Ian Geisler M. Fernandez

Course & Section: IT 114- AF1

Title: Understanding ReactJS Props and State

Objective:

The objective of this lab activity is to help you understand the concepts of ReactJS props and state and how they can be used to create dynamic and interactive UIs.

Requirements:

• Basic knowledge of HTML, CSS, and JavaScript

• Install Node.js and create a new ReactJS project and upload your code in an online repository such as GitHub (https://github.com/) or create your code directly in your CodeSanbox.io account

What to Submit:

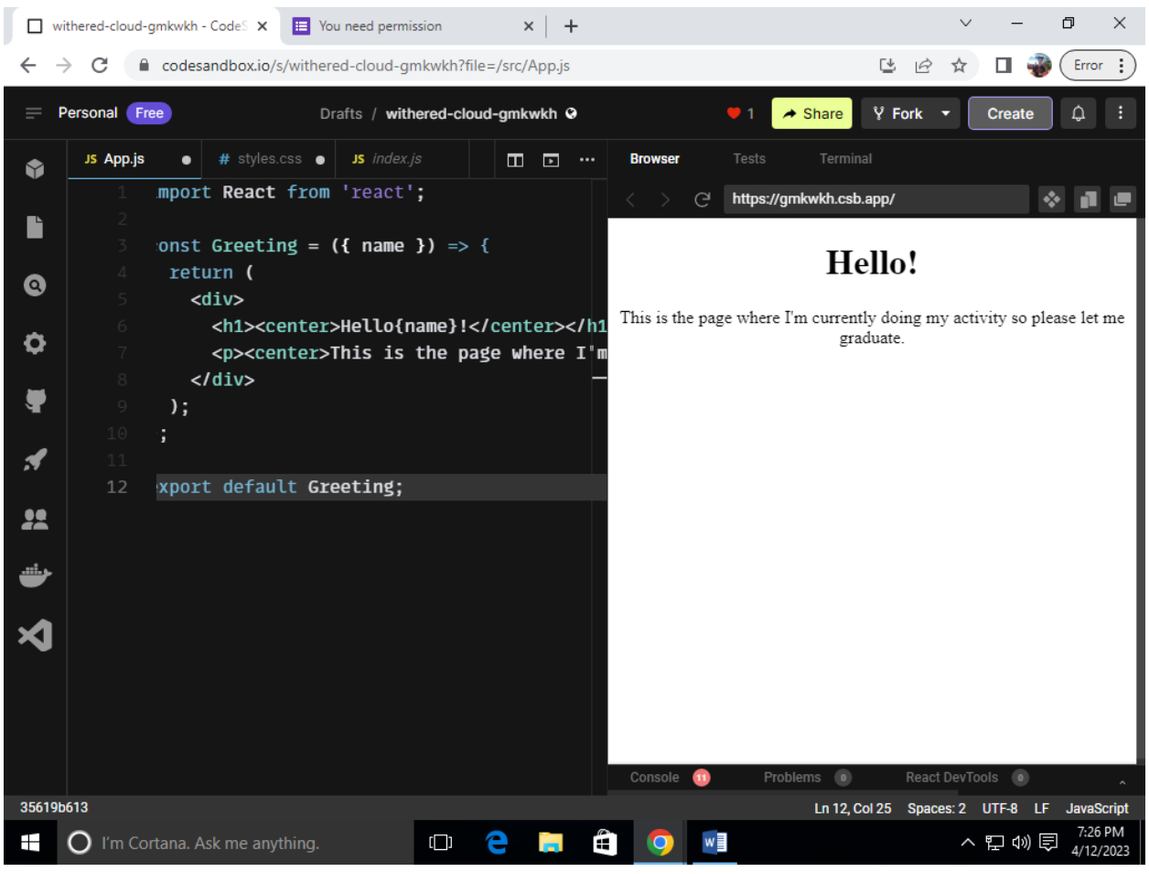
• GitHub/CodeSanbox link containing your code.

• Submit it here: https://forms.gle/Mm7kfzweP6fbZAwG9

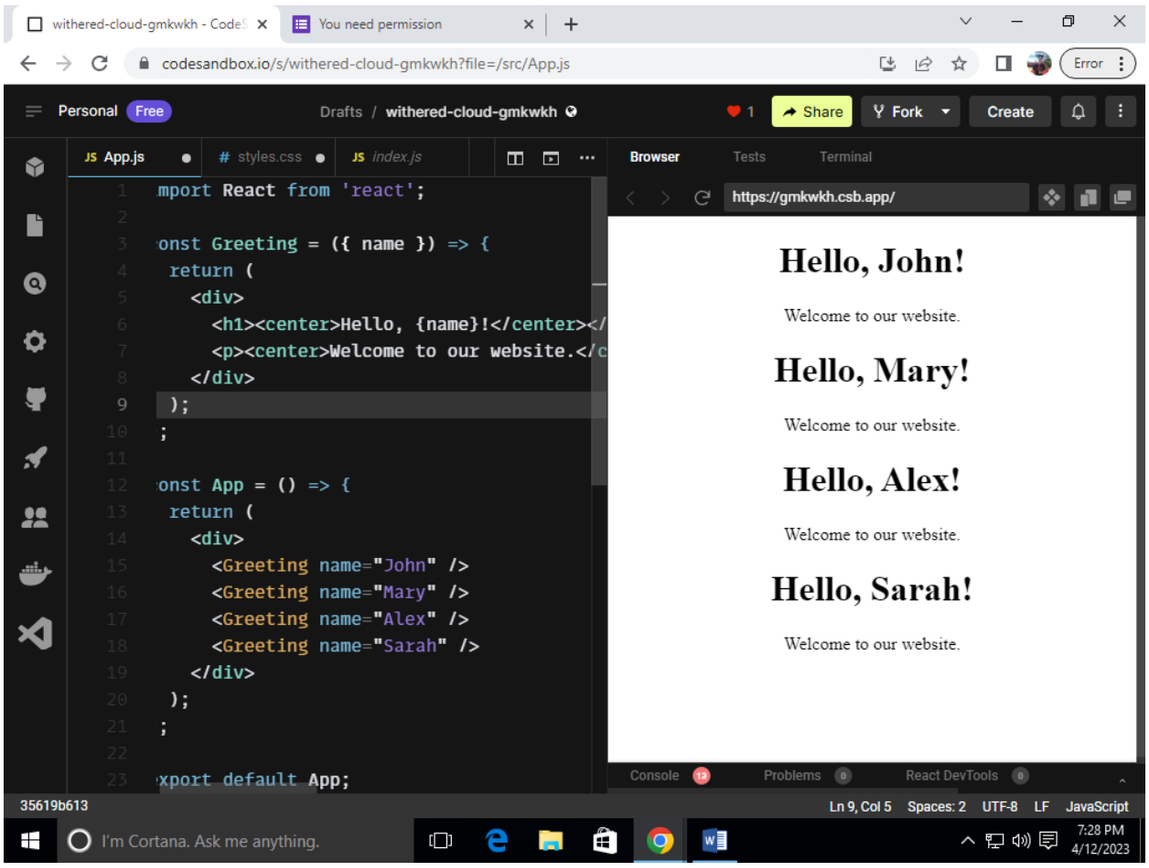
• Deadline of submission is on April 12, 2023

Task 1: Understanding Props

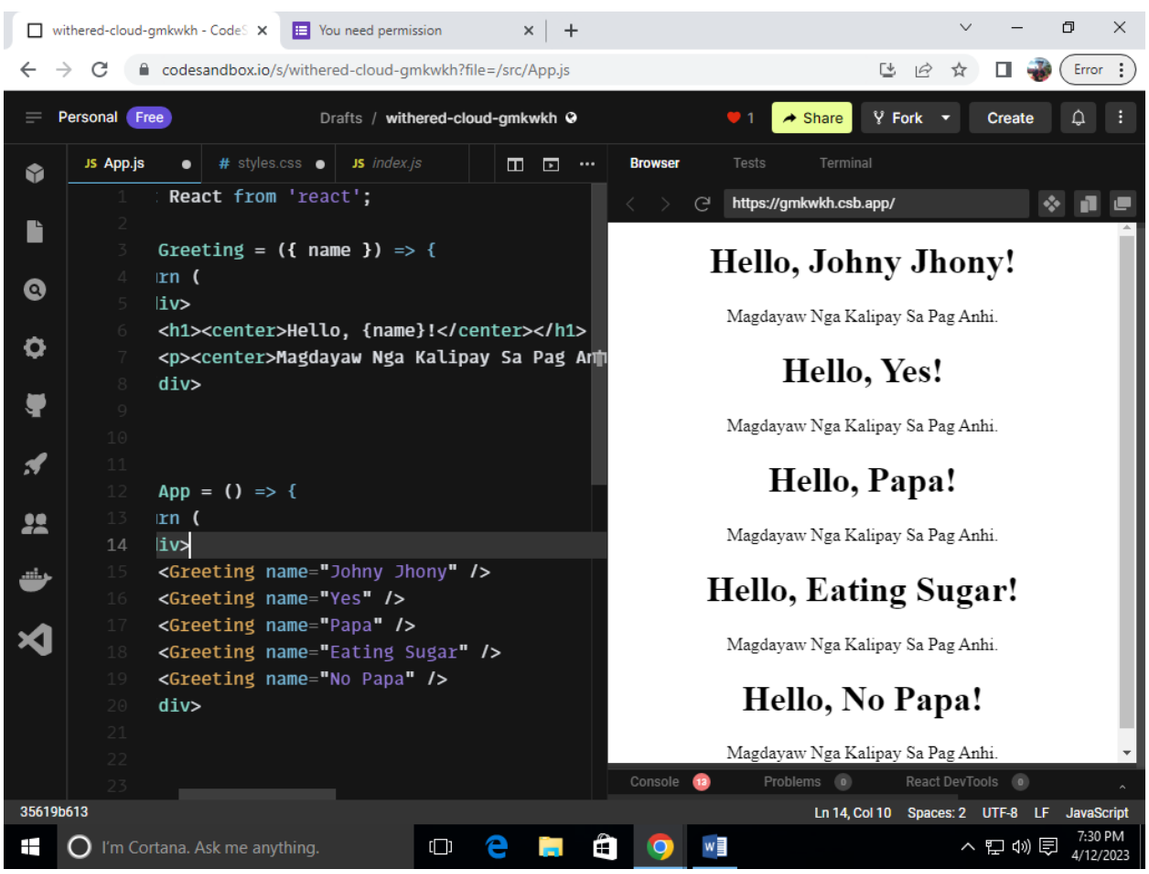
1. Create a new ReactJS component called Greeting that takes a name prop and displays a personalized greeting message.



2. In the App component, render the Greeting component with different name props, such as "John", "Mary", "Alex", and "Sarah".

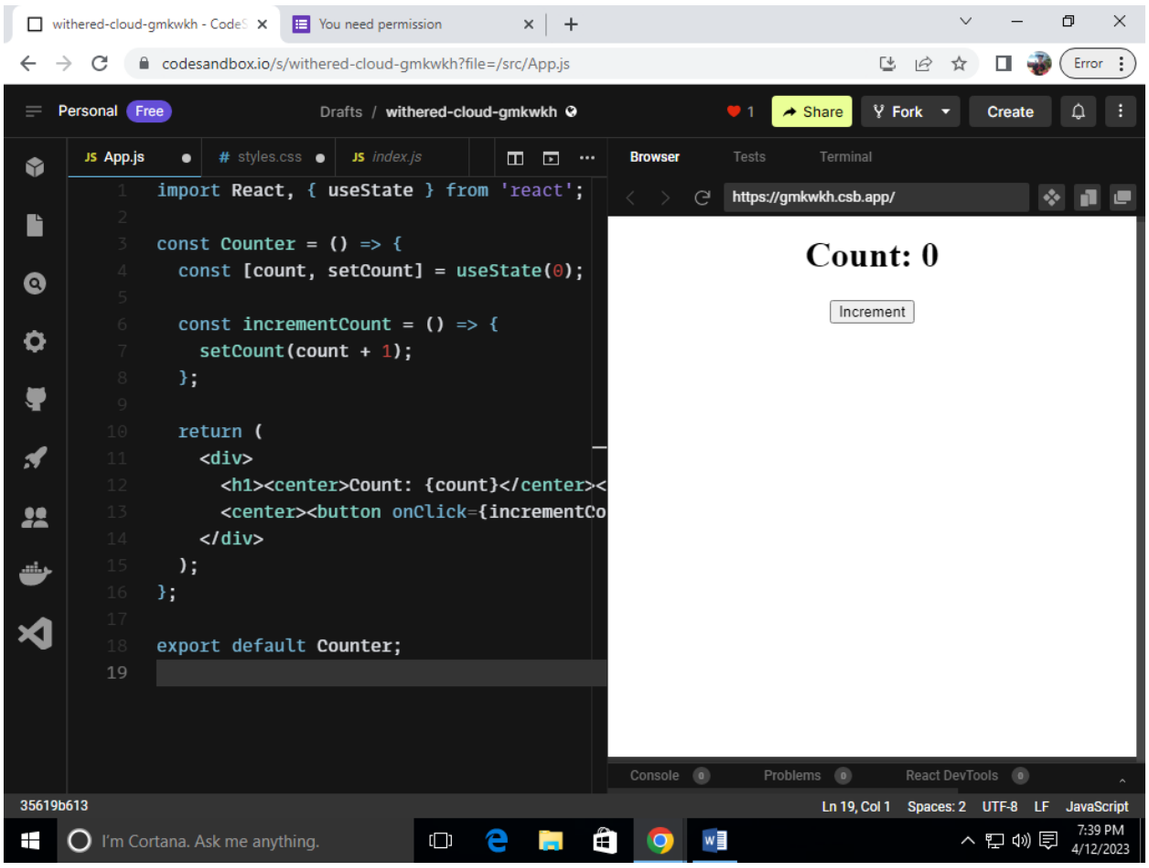


3. Observe how the Greeting component updates the message based on the name prop passed to it.

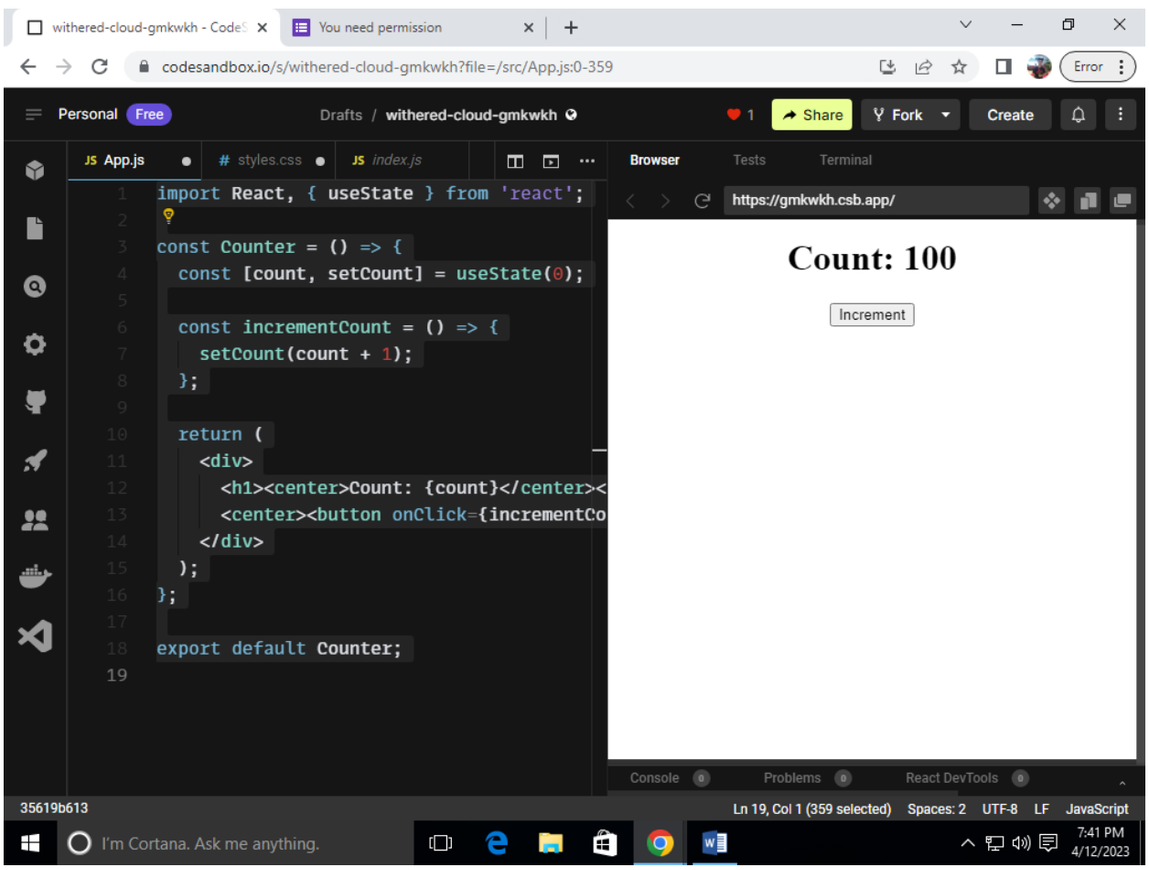


Task 2: Understanding State

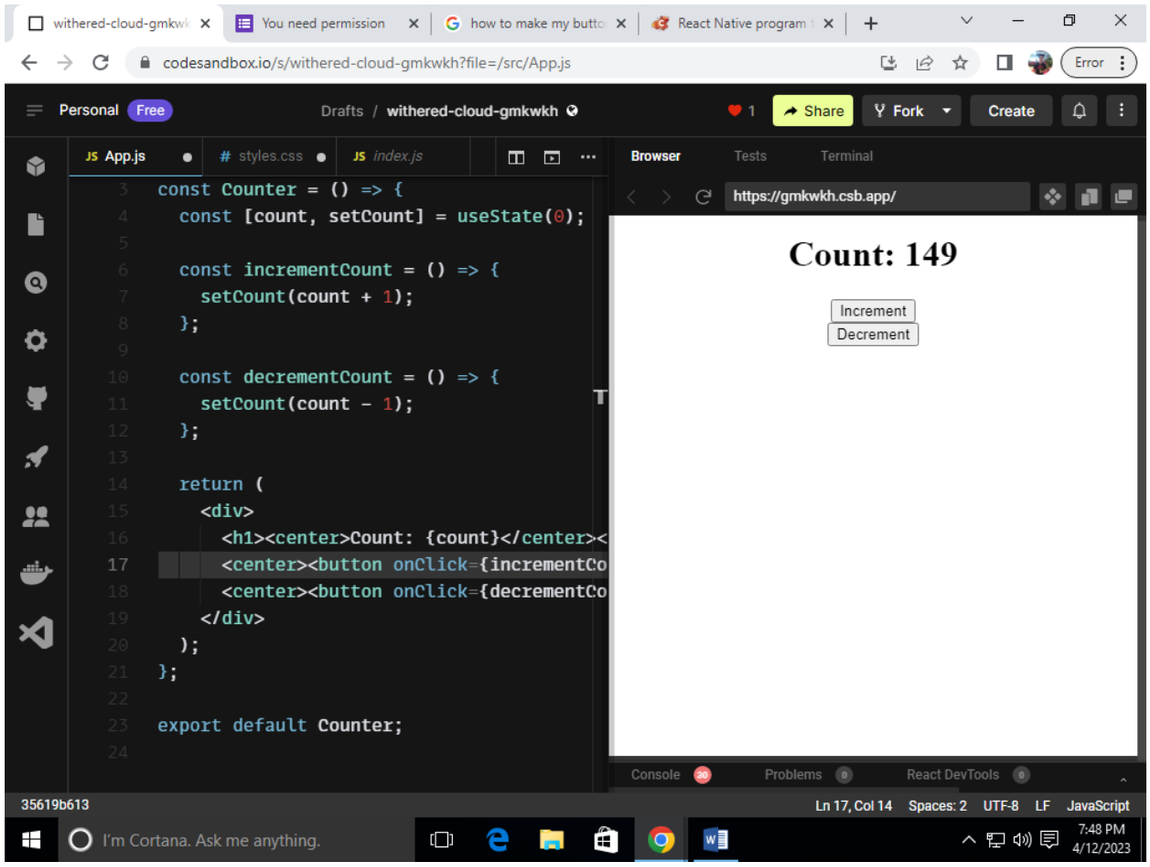
1. Create a new ReactJS component called Counter that has a count state initialized to 0 and displays a button to increment the count.



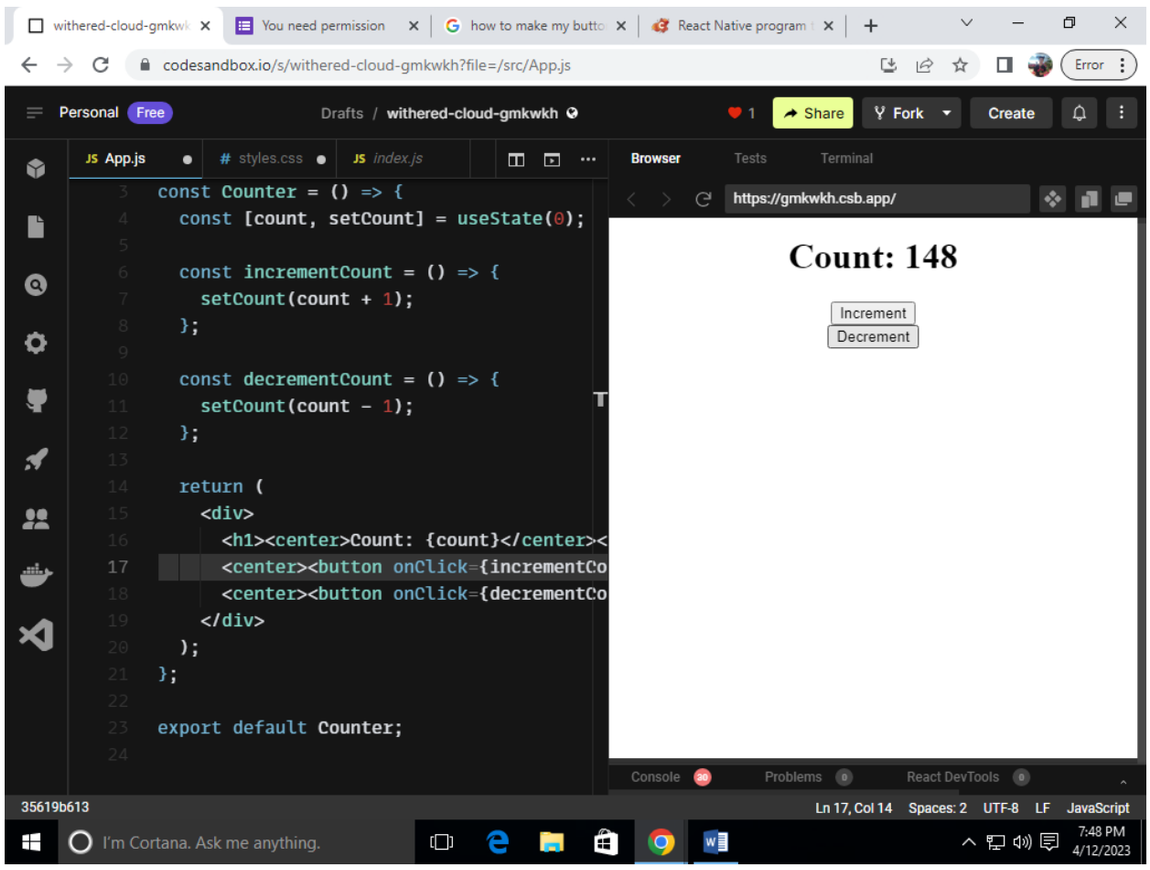
2. Implement a click event handler for the button that updates the count state by 1.



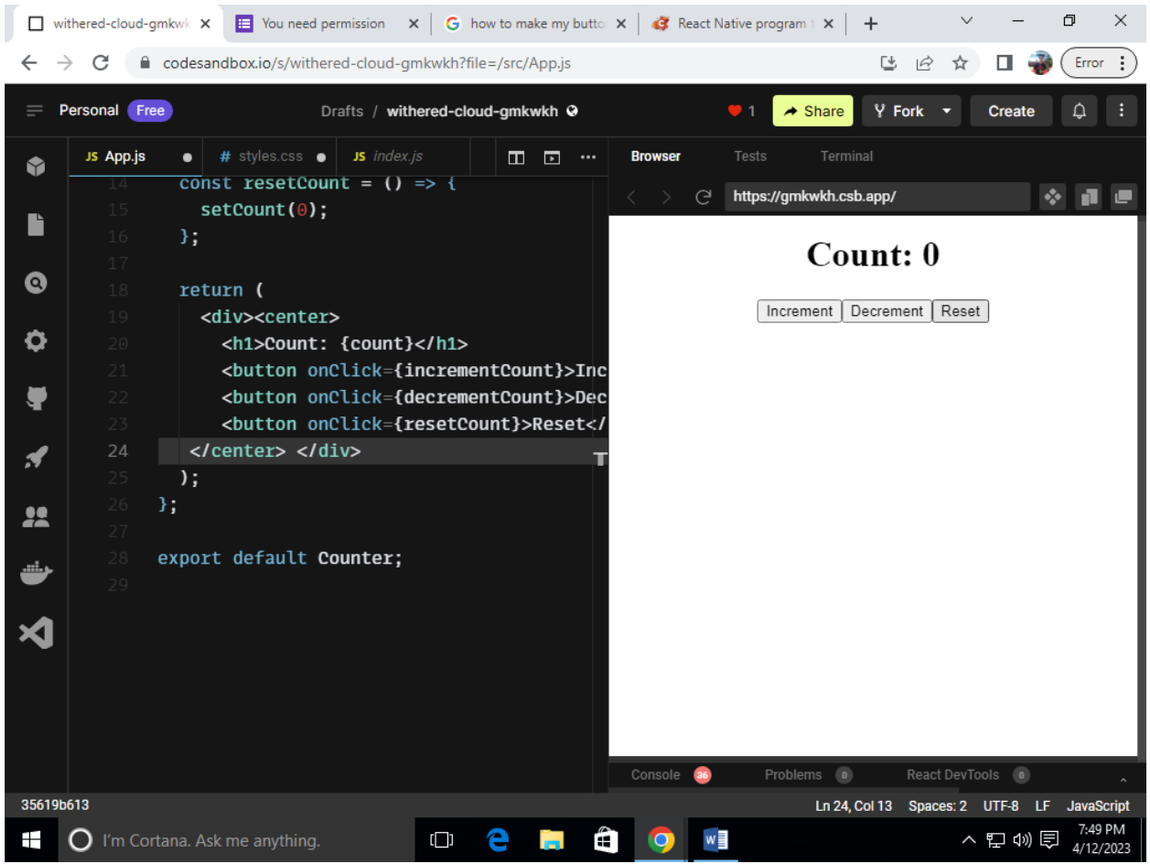
3. In the App component, render the Counter component and observe how the count state changes when the button is clicked.



4. Implement a button to decrement the count state by 1.

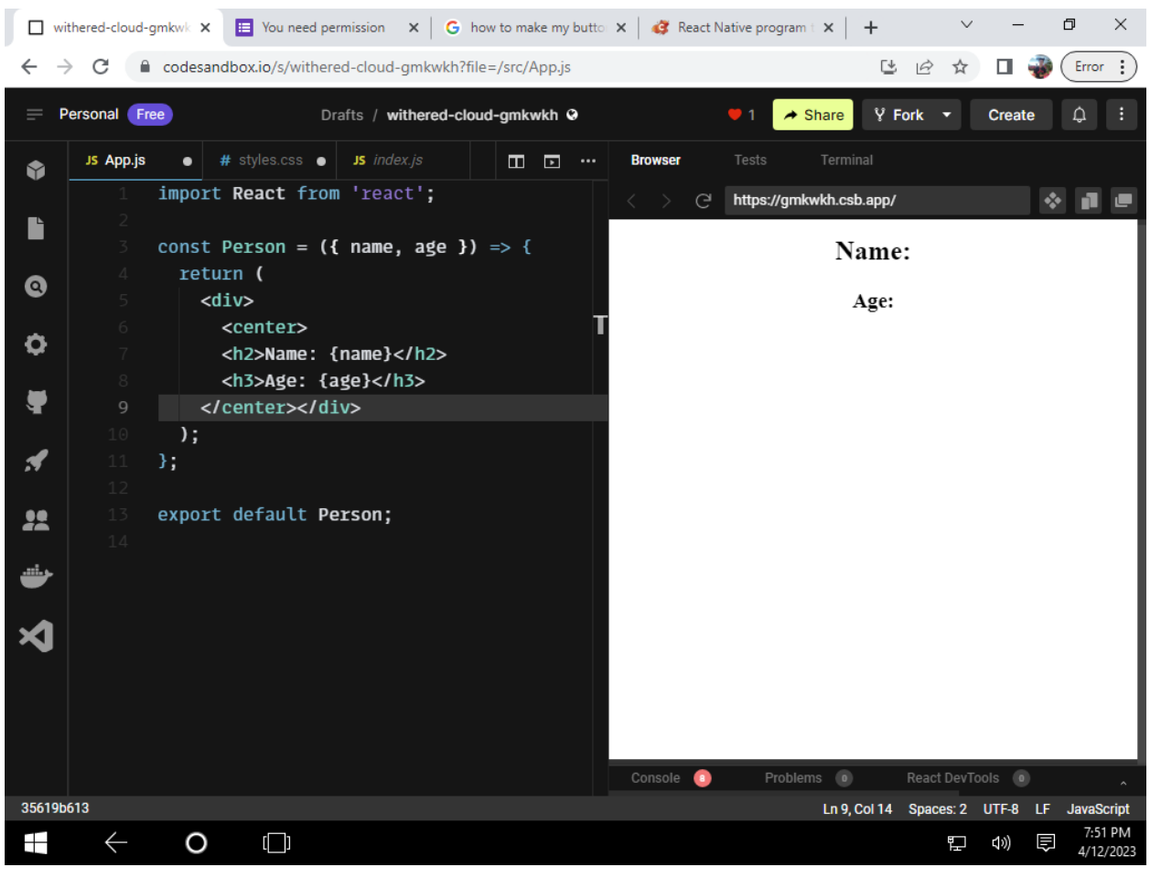


5. Implement a button to reset the count state to 0.

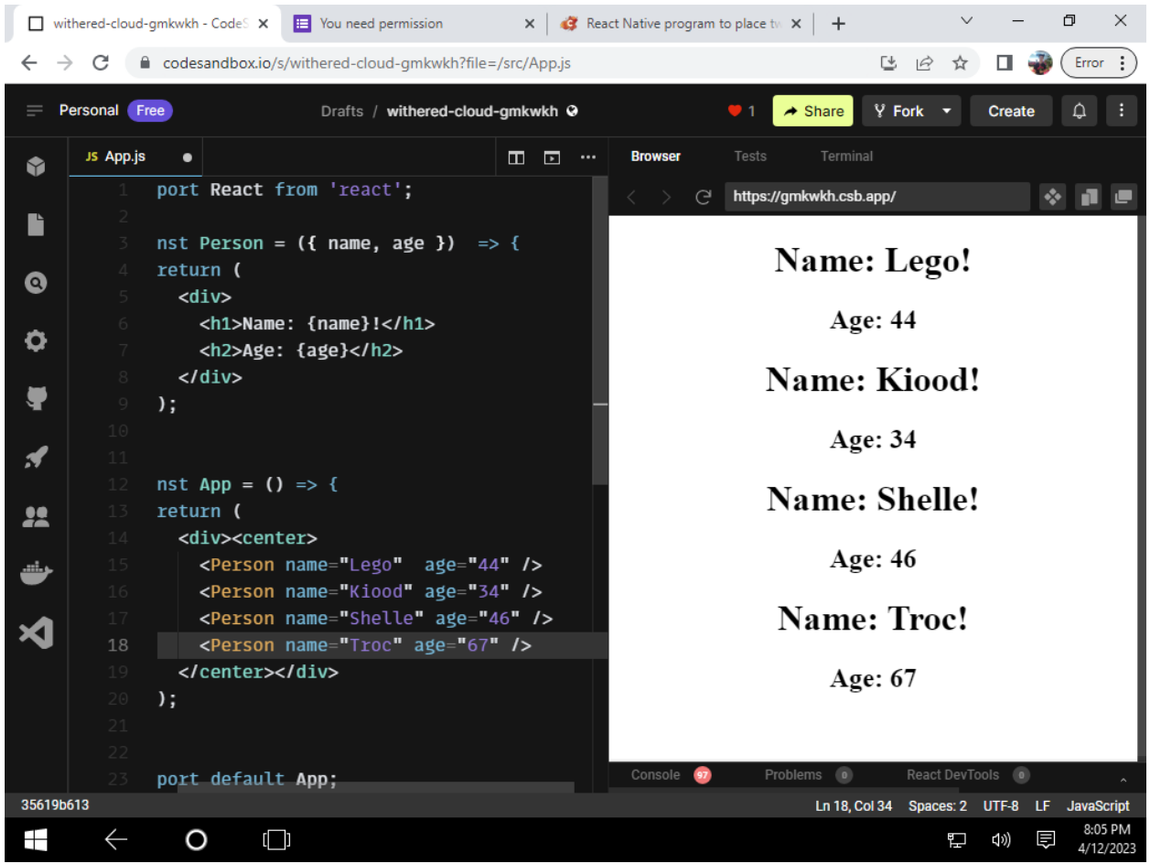


Task 3: Combining Props and State

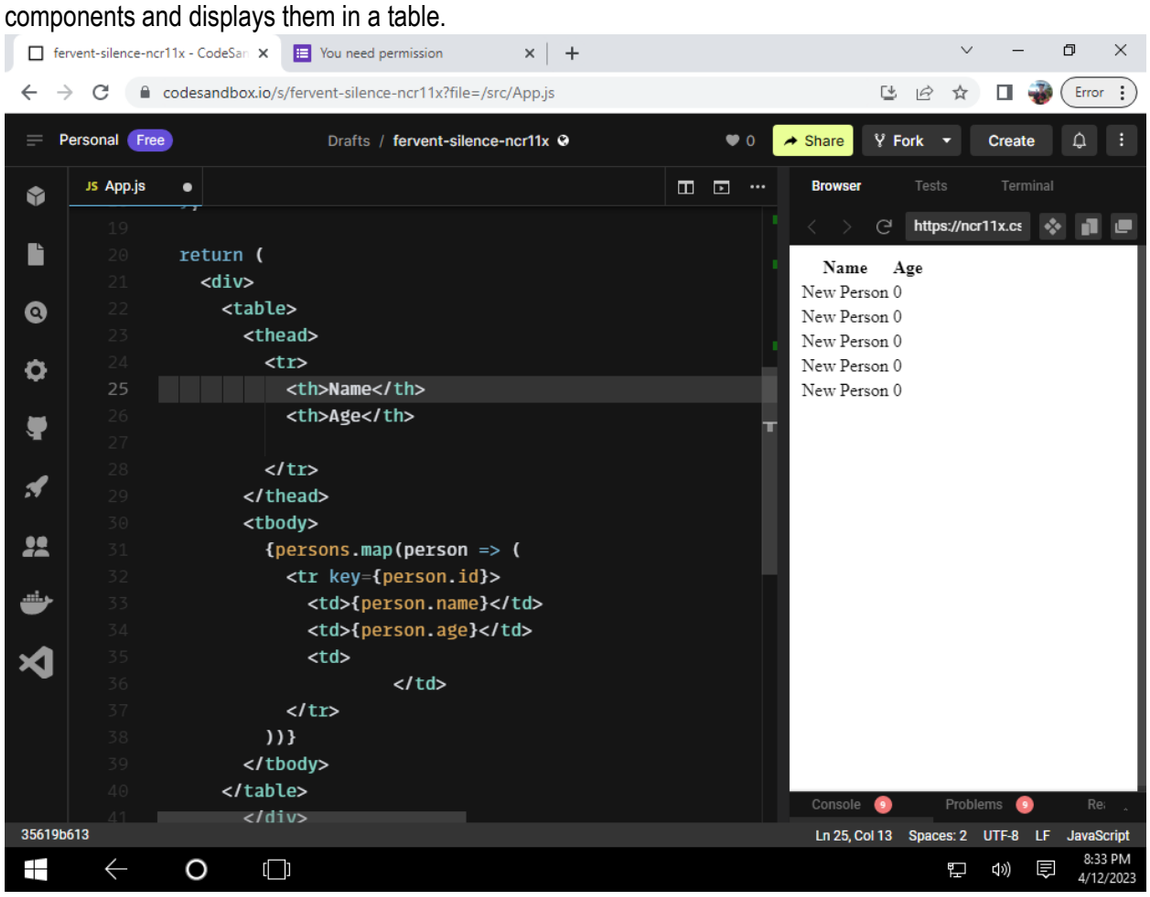
1. Create a new ReactJS component called Person that takes a name prop and an age prop, and displays the person's name and age.



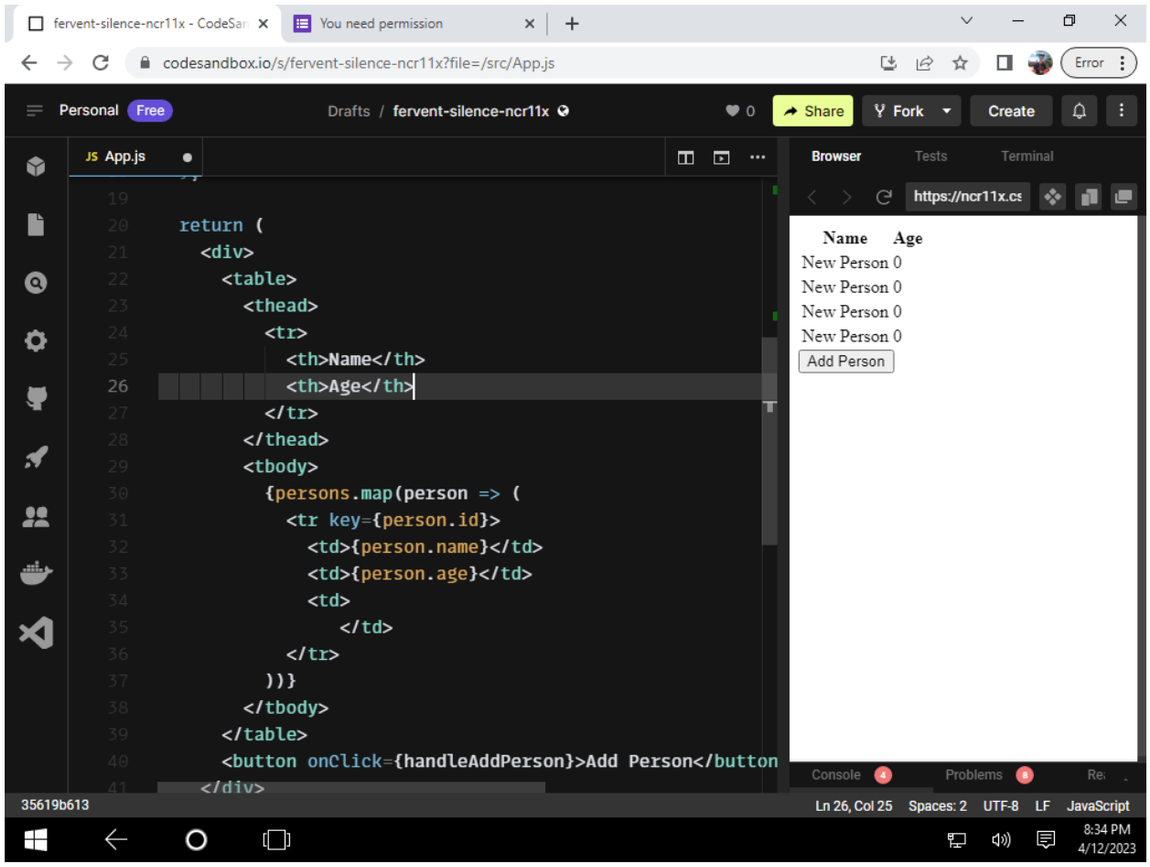
2. In the App component, render the Person component with different name and age props.



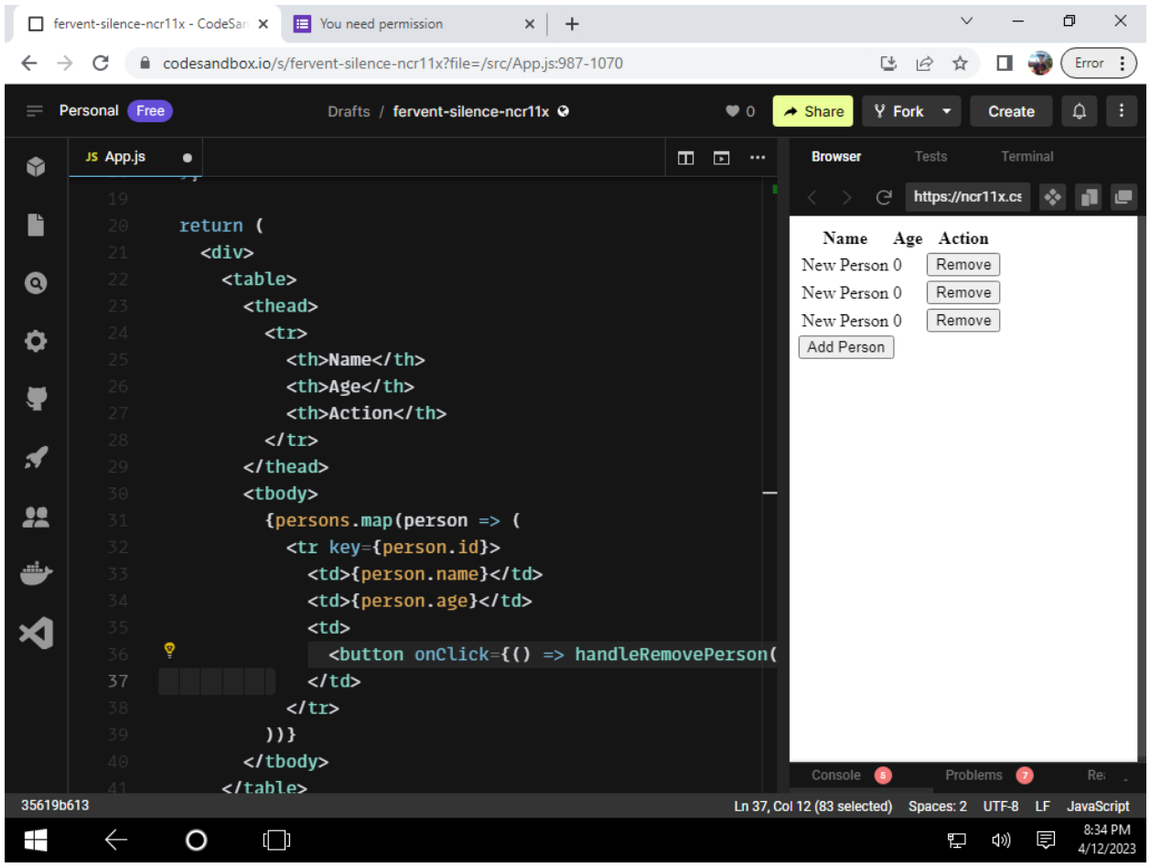
3. Create a new ReactJS component called PersonList that maintains a list of Person components and displays them in a table.



4. Implement a button to add a new Person component to the PersonList component.



5. Implement a button to remove a Person component from the PersonList component.



Conclusion:

In this lab activity, you learned about the concepts of ReactJS props and state, and how they can be used to create dynamic and interactive UIs. By combining these concepts, you can create complex components that can be reused throughout your application.