**Lists and Keys**

Q1: How do you render a list of items in React? Why is it important to use keys when rendering lists?

Ans: - React render a list by mapping over an array and returning JSX elements for each item. **Keys** are special string attributes you must include when creating lists of elements in React.

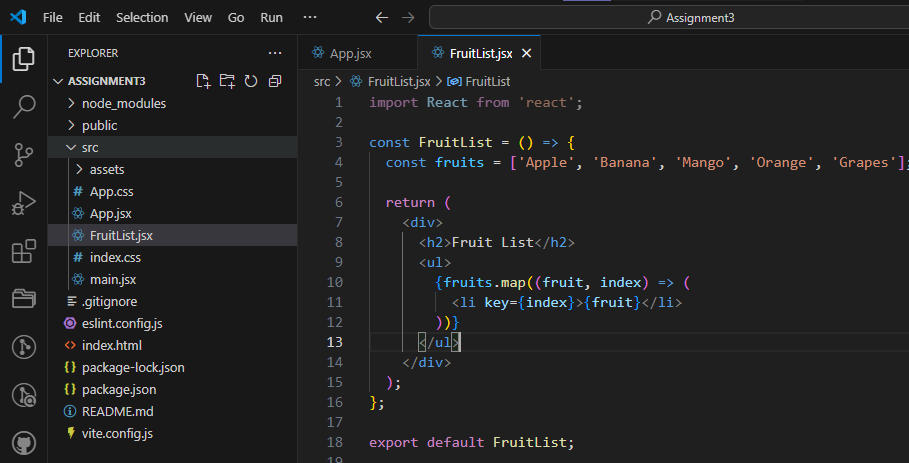
Q2: What are keys in React, and what happens if you do not provide a unique key?

Ans: - Keys in React are unique identifiers assigned to elements in a list to help React efficiently track and update individual items during rendering. It can lead to unexpected behavior, like losing input focus or component state.

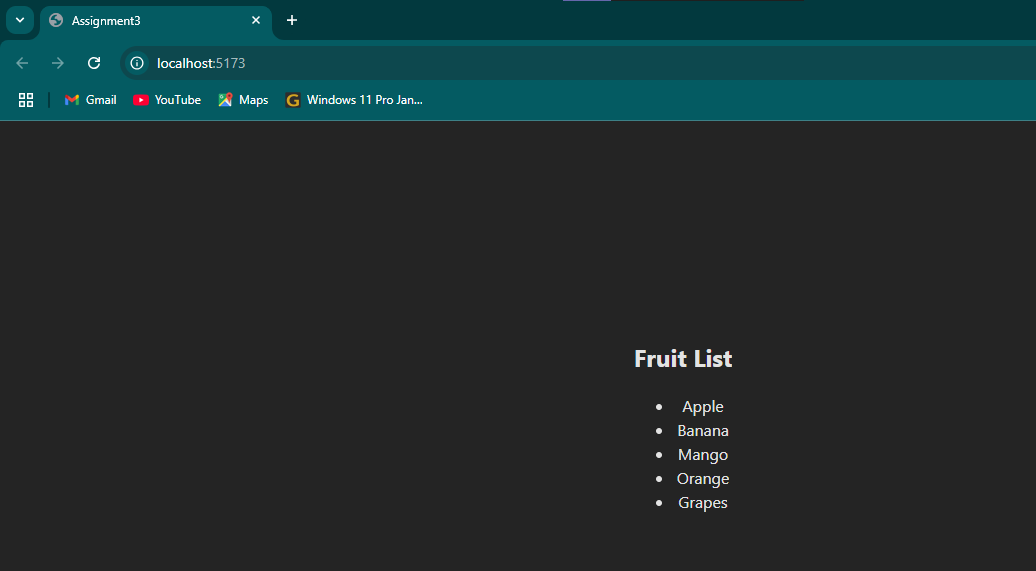
**Lab Tasks**

**Task 1 : Create a React component that renders a list of items (e.g., a list of fruit names). Use the map() function to render each item in the list.**

**Code:**

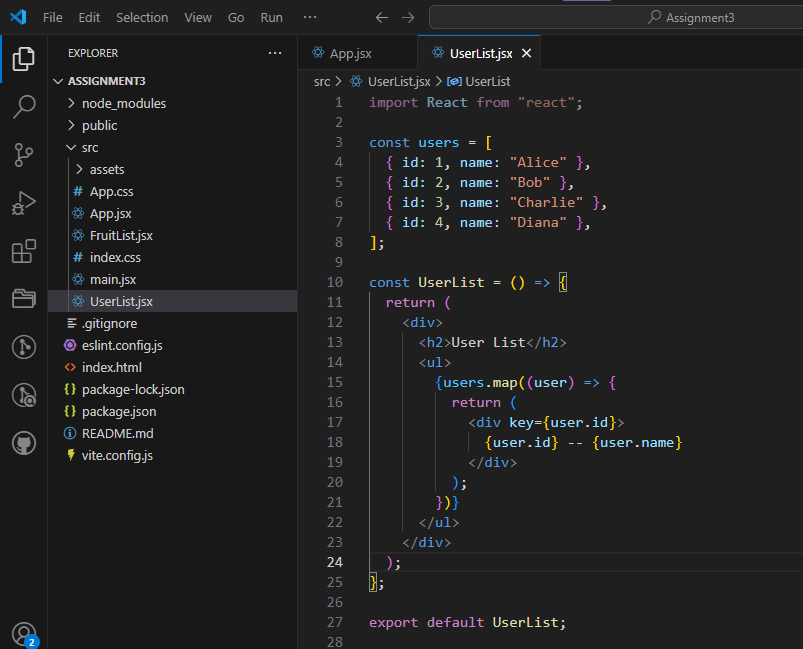
****

**Output:**

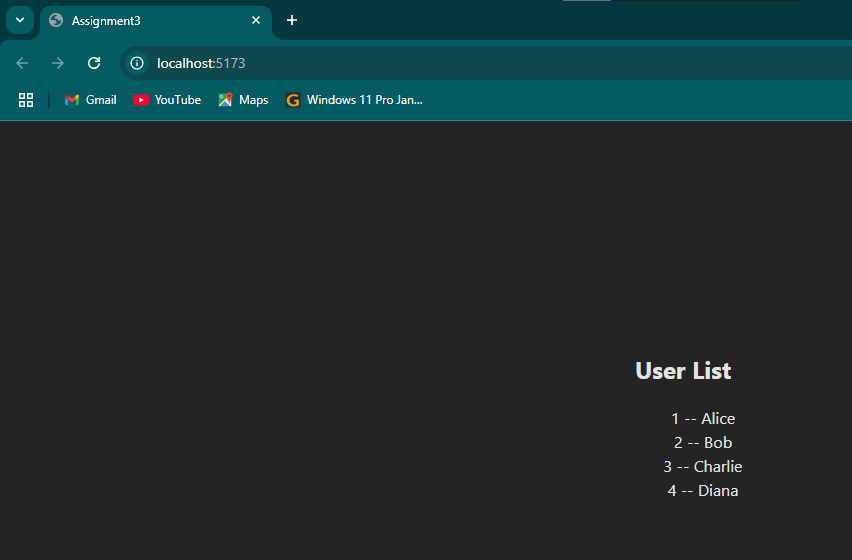
****

**Task 2 : Create a list of users where each user has a unique id. Render the user list using React and assign a unique key to each user.**

**Code:**

****

**Output:**

****

**Hooks (useState, useEffect, useReducer, useMemo, useRef, useCallback)**

Q1: What are React hooks? How do useState() and useEffect() hooks work in functional components?

Ans: - React Hooks are special functions introduced in React 16.8 that allow you to use state and other React features in functional components.

1. **useState()** :- is a React Hook that allows you to add state to functional components. It returns a stateful value and a function to update it.
2. **useEffect()** :- is a React Hook that lets you perform side effects in functional components. It runs after the component renders and can also clean up after itself.

Q2: What problems did hooks solve in React development? Why are hooks considered an important addition to React?

Ans: - Hooks were added to React to fix some problems with using class components. Before hooks, it was hard to share code between components, manage complicated lifecycle methods, and there was a lot of extra code to write. Hooks let you use state and other features inside simple functions instead of classes. This makes the code easier to write, understand, and reuse. Hooks are important because they make React simpler and help developers build cleaner and better apps.

Q3: What is useReducer ? How we use in react app?

Ans: - useReducer is a React Hook that helps you manage complex state logic in a component. It’s an alternative to useState when you have state that depends on previous state or involves multiple related values. It works similar to Redux reducers but is built into React.

Q4: What is useReducer ? How we use in react app?

Ans: - useReducer is a React Hook that helps you manage complex state logic in a component. It’s an alternative to useState when you have state that depends on previous state or involves multiple related values. It works similar to Redux reducers but is built into React.

Q5: What’s the Difference between the useCallback & useMemo Hooks?

Ans: -

* **useCallback** **--** memoizes a function. It returns a memoized version of a callback function that only changes if its dependencies change.
* **useMemo** -- memoizes a value. It returns a memoized result of a calculation that only recomputes when its dependencies change.

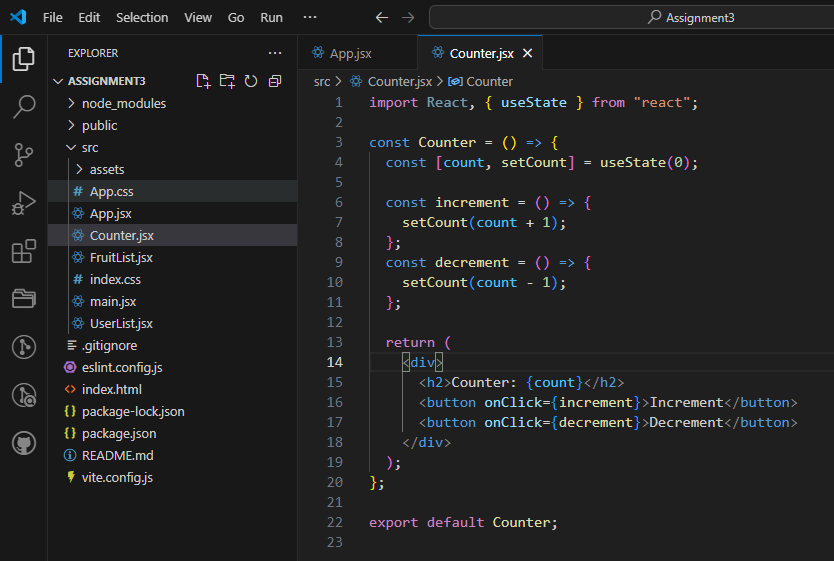
Q6: What is useRef ? How to work in react app?

Ans: - useRef is a React Hook that gives you a way to create a reference to a DOM element or to store a value that doesn’t re-render the component when it changes.

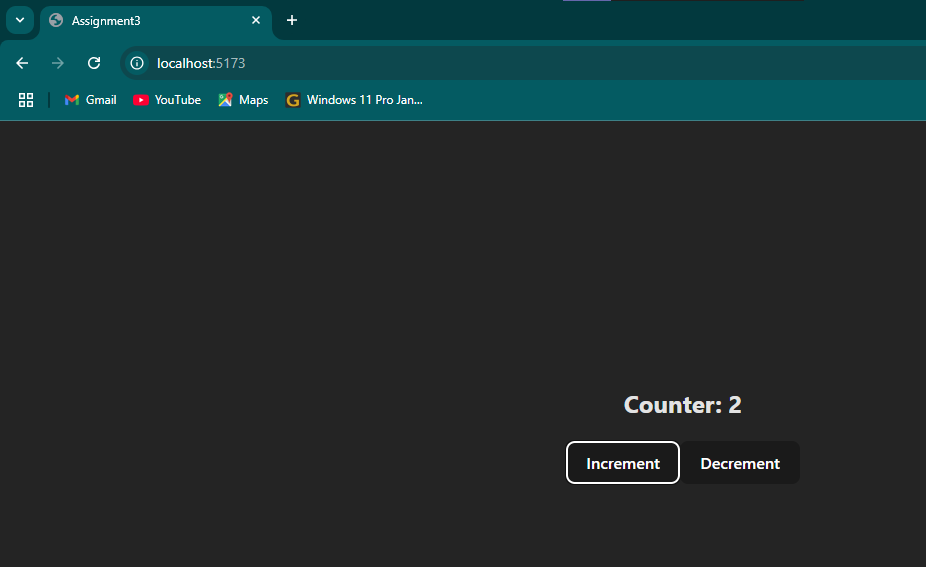
**Lab Tasks**

**Task 1 : Create a functional component with a counter using the useState() hook. Include buttons to increment and decrement the counter.**

**Code:**

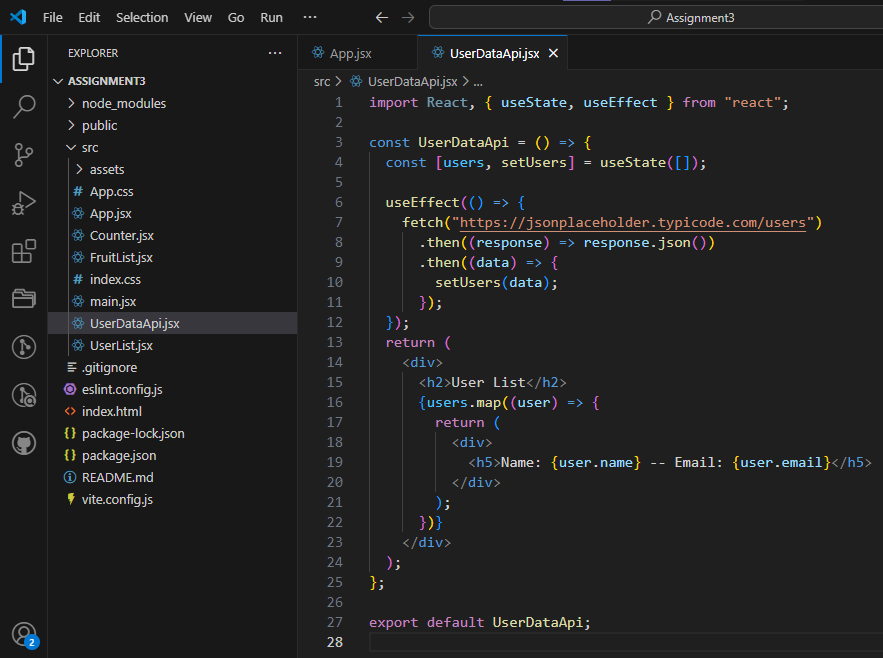


**Output:**

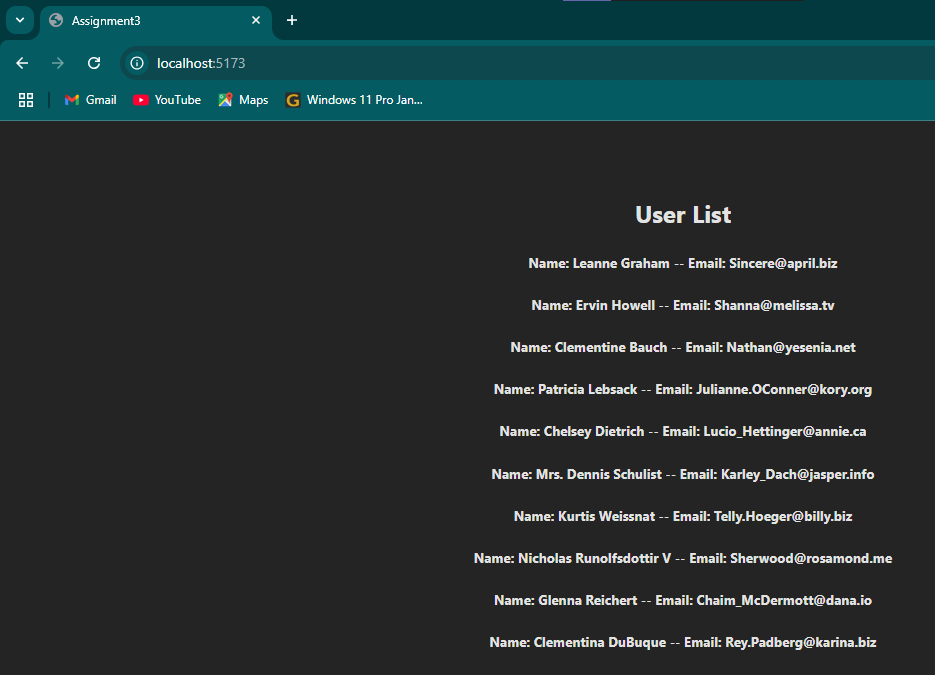
****

**Task 2 : Use the useEffect() hook to fetch and display data from an API when the component mounts.**

**Code:**

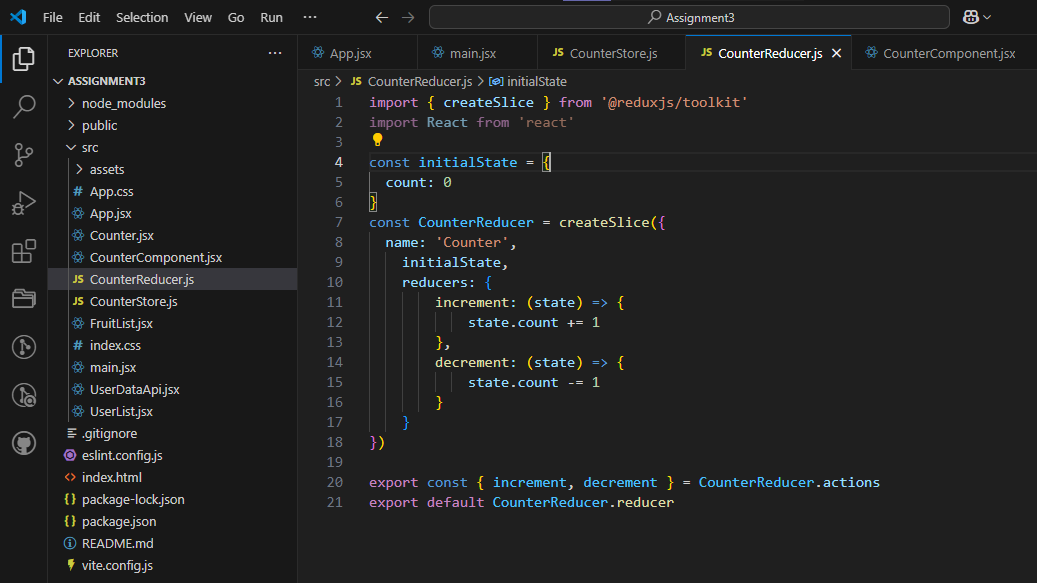
****

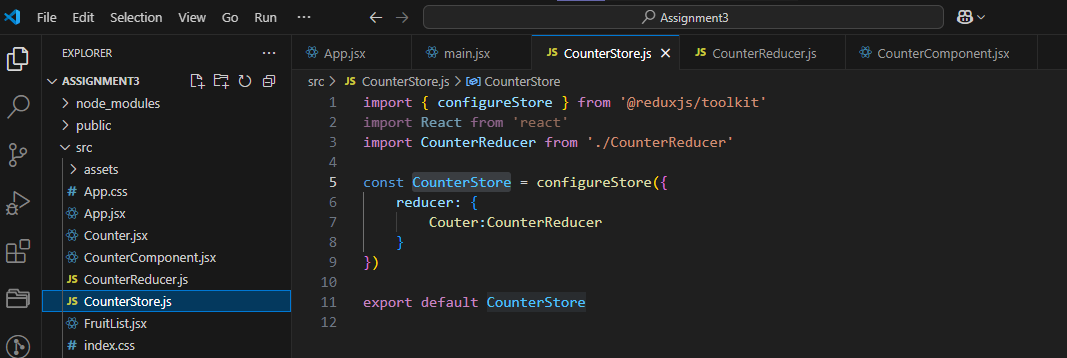
**Output:**

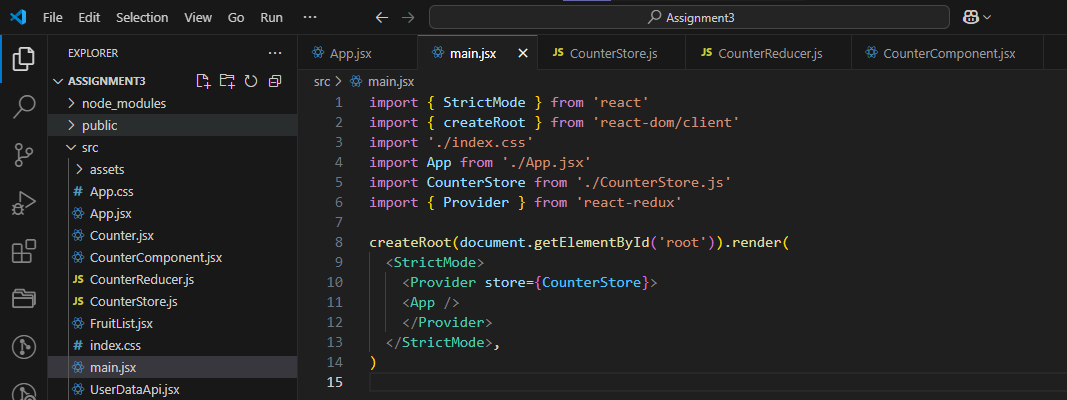
****

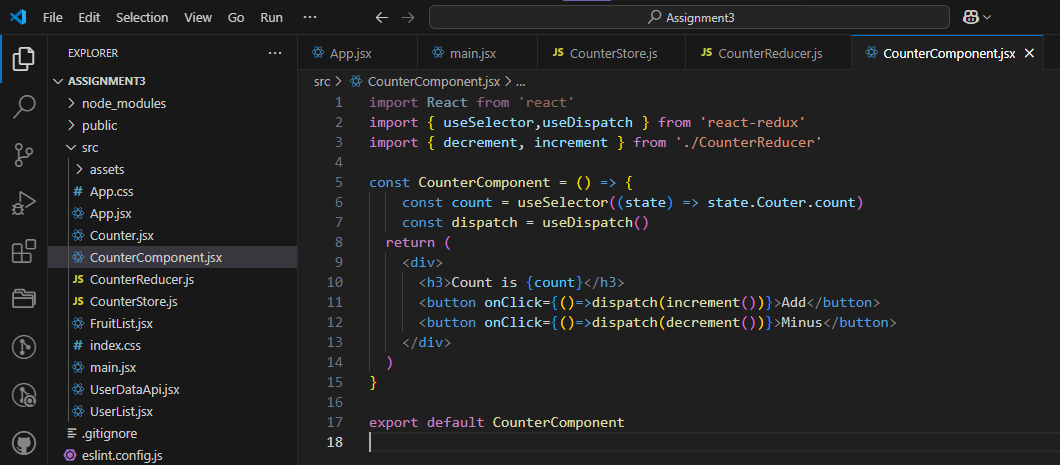
**Task 3 : Create react app with use of useSelector & useDispatch.**

**Code:**

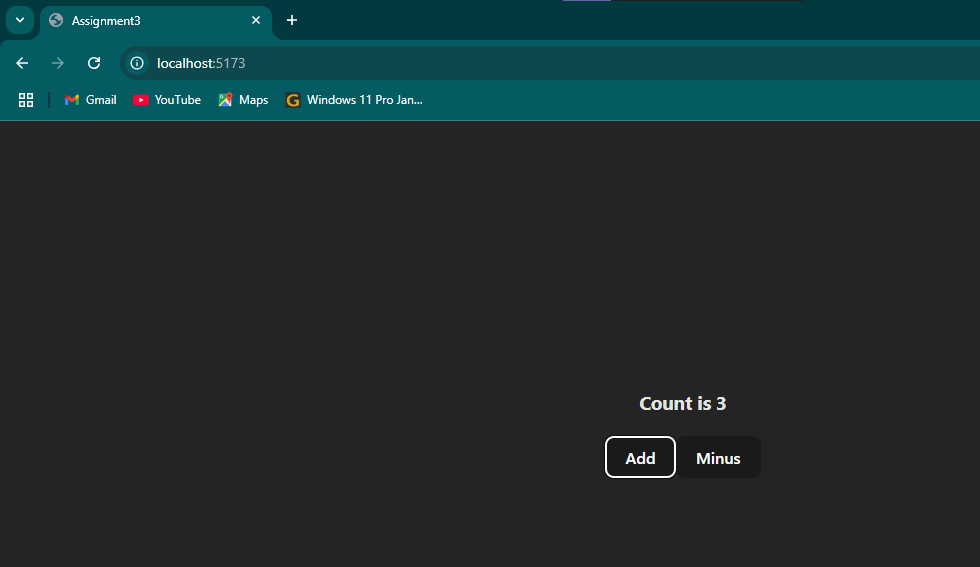
****

****

****

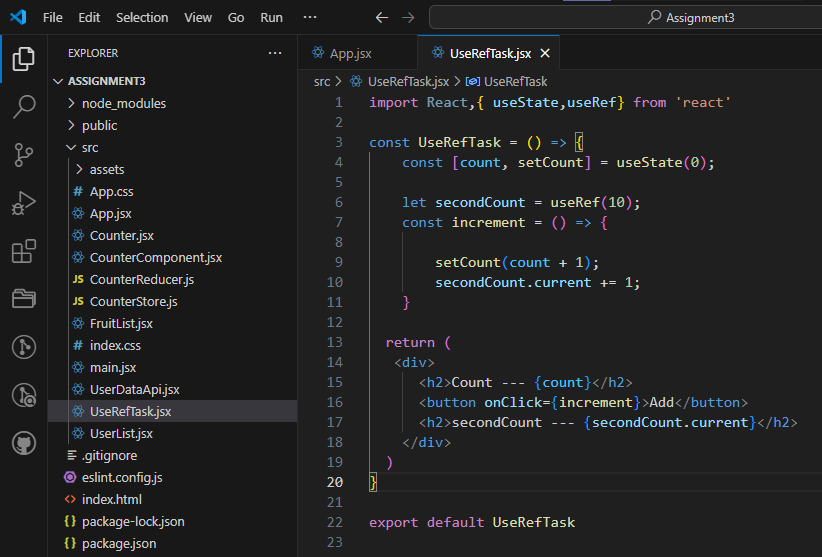
****

**Output:**

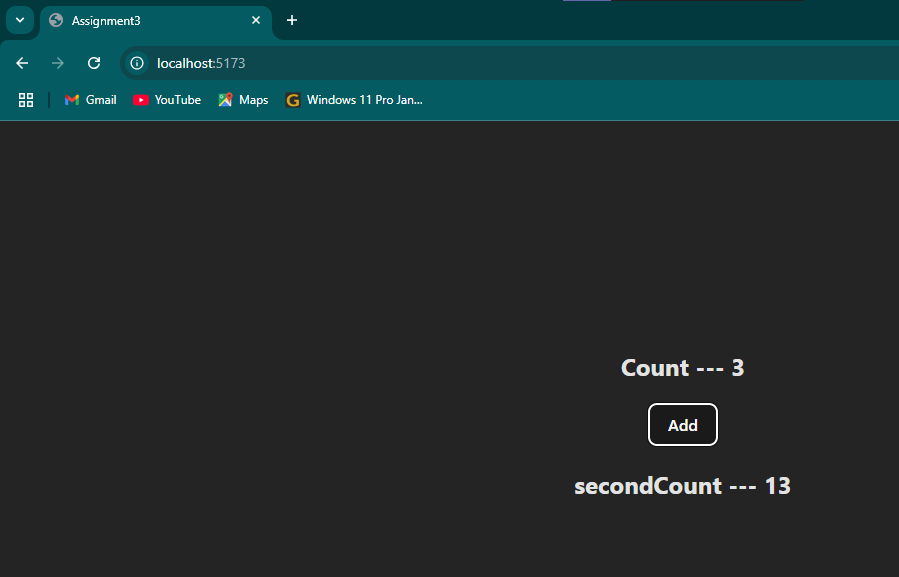
****

**Task 4 : Create react app with use of useSelector & useDispatch.**

**Code:**

****

**Output:**

****