Lesson 04 Demo 01

Creating a React Application Using the useReducer Hook

Objective: To develop a React application that demonstrates the useReducer Hook

Tools required: Node terminal, React app, and Visual Studio Code

Prerequisites: Knowledge of creating a React app and understanding of the folder structure

Steps to be followed:

1. Create a new React app
2. Create a new reducer function in App**.**js
3. Initialize state using the useReducer Hook in App.js
4. Run the app and view it in the browser

Step 1: Create a new React app

1.1 Run the **npx create-react-app use-reducer-demo** command in the terminal to create a
new **React** app in a directory named **use-reducer-demo**

1.2 Run the **cd use-reducer-demo** command in the terminal to move into the project directory

Step 2: Create a new reducer function in App.js

2.1 Open the **Visual Studio Code** and navigate to the **use-reducer-demo** directory
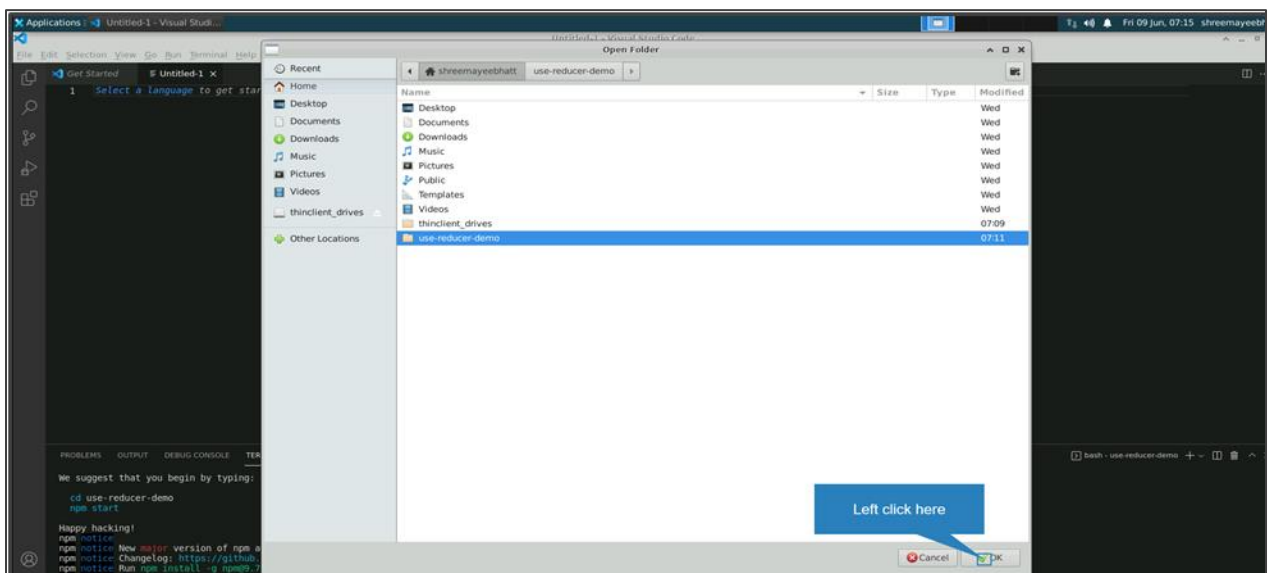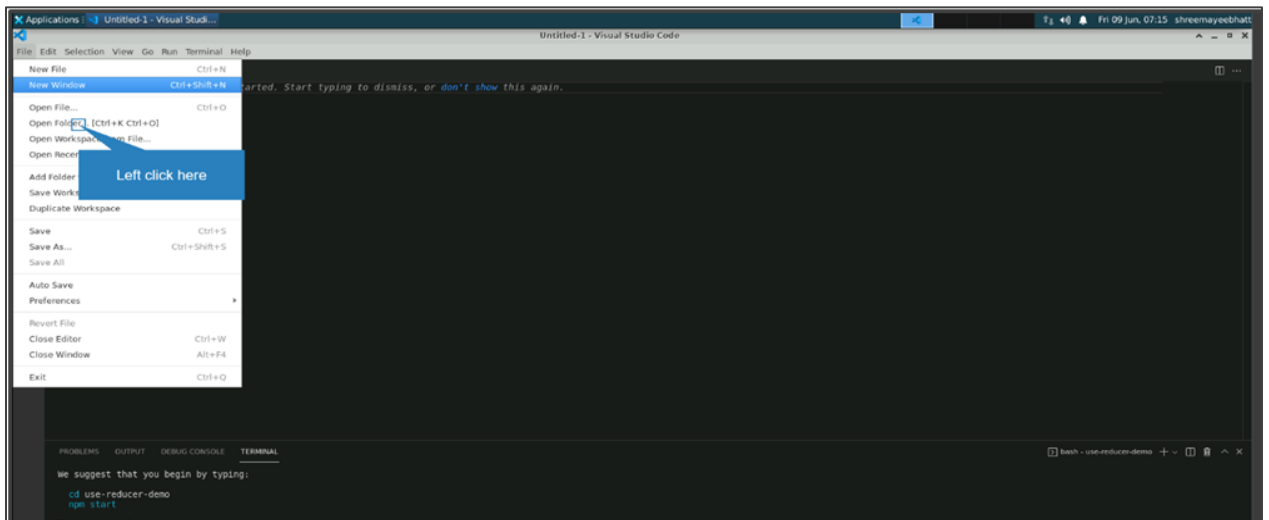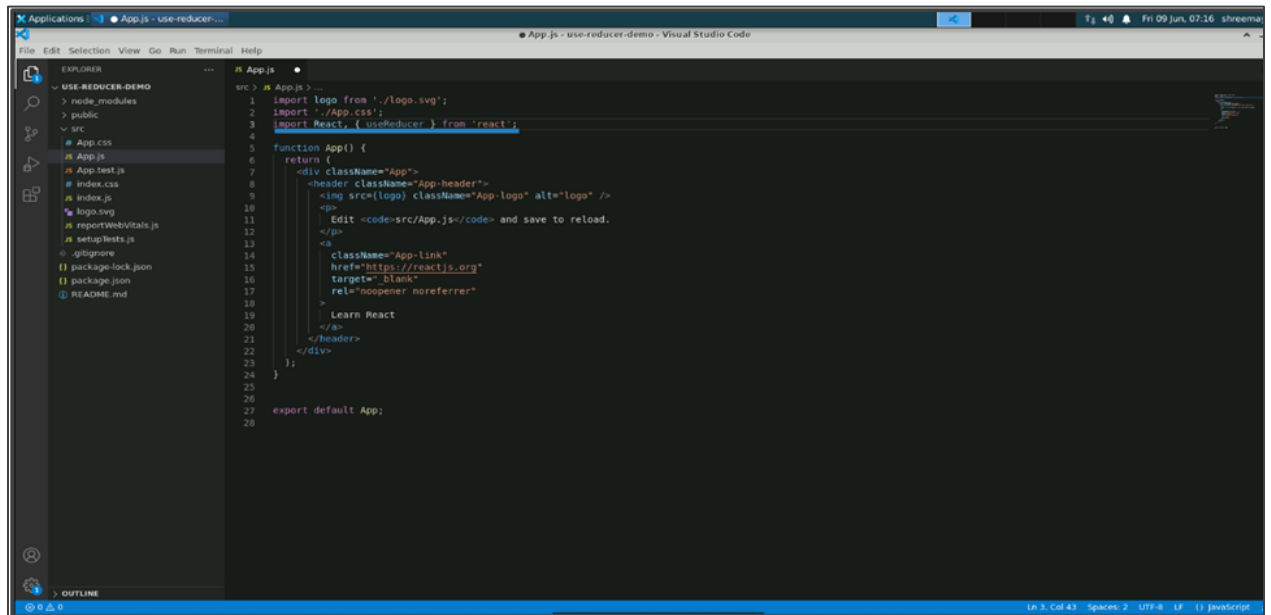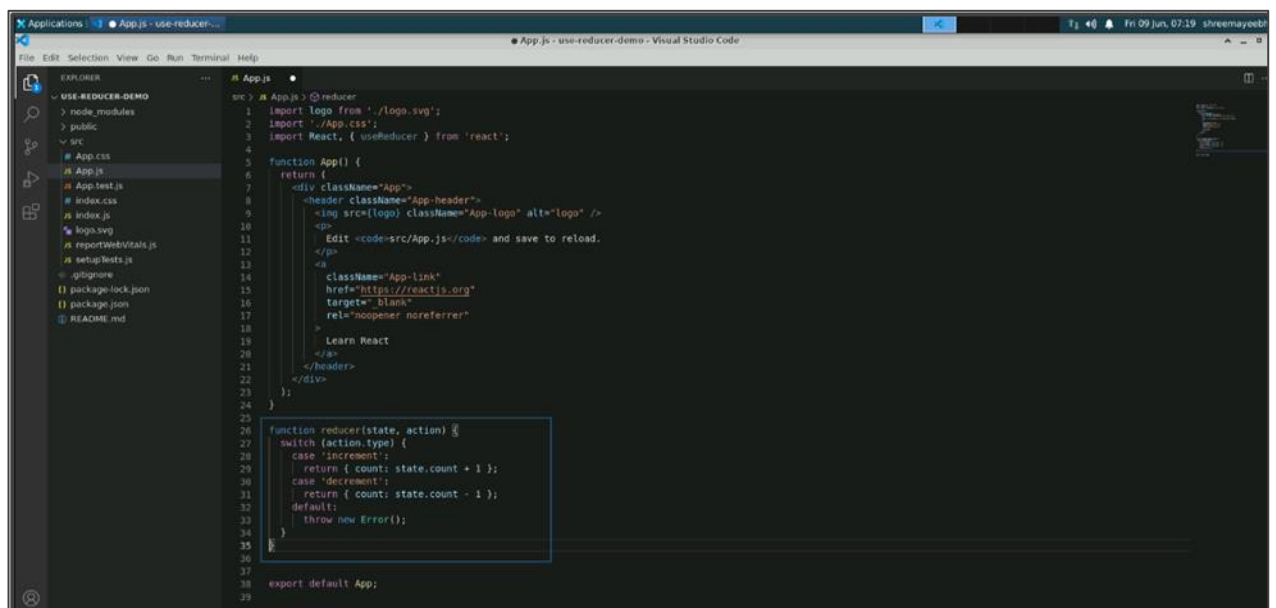
2.2 Inside the project directory, navigate to **src/App.js** and import **React** and **useReducer** from the React library using the following command:

import React, { useReducer } from React;

2.3 Define a function named **reducer** to create a simple counter that can be incremented or decremented

```
function reducer(state, action) {
 switch (action.type) {
  case 'increment':
    return { count: state.count + 1 };
  case 'decrement':
    return { count: state.count - 1 };
  default:
   throw new Error();
 }
}
```

Step 3: Initialize state using the useReducer Hook in App.js

3.1 In the **App.js** file, inside the **App** function, pass the **reducer** function and an initial state object to the **useReducer** Hook and assign the returned **state** and **dispatch** function to **state** and **dispatch** variables.

```
function App() {
  const [state, dispatch] = useReducer(reducer, { count: 0 });
```

3.2 Finally, configure the **App()** function by entering the following code:

```
function App() {
const [state, dispatch] = useReducer(reducer, { count: 0 });
return (
<div>
    <h1>useReducer Demo</h1>
    <p>Count: {state.count}</p>
    <button onClick={() => dispatch({ type: 'increment' })}>+</button>
    <button onClick={() => dispatch({ type: 'decrement' })}>-</button>
</div>
);
}
export default App;
```

```
JS App.js          ●

src > JS App.js > ⊗ App
  1    import logo from './logo.svg';
  2    import './App.css';
  3    import React, { useReducer } from 'react';
  4
  5    function App() {
  6      const [state, dispatch] = useReducer(reducer, { count: 0 });
  7
  8      return (
  9        <div>
 10          <h1>useReducer Demo</h1>
 11          <p>Count: {state.count}</p>
 12          <button onClick={() => dispatch({ type: 'increment' })}>+</button>
 13          <button onClick={() => dispatch({ type: 'decrement' })}>-</button>
 14        </div>
 15
 16      );
 17    }
 18
 19    function reducer(state, action) {
 20      switch (action.type) {
 21        case 'increment':
 22          return { count: state.count + 1 };
 23        case 'decrement':
 24          return { count: state.count - 1 };
 25        default:
 26          throw new Error();
 27      }
 28    }
 29
 30
 31    export default App;
 32
```

Step 4: Run the app and view it in the browser

4.1 In the terminal, run the **npm start** command to start the app

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

shreemayeebhatt@ip-172-31-22-250:~/use-reducer-demo$ npm start
```

4.2 To view the output, open the browser and navigate to **http://localhost:3000**
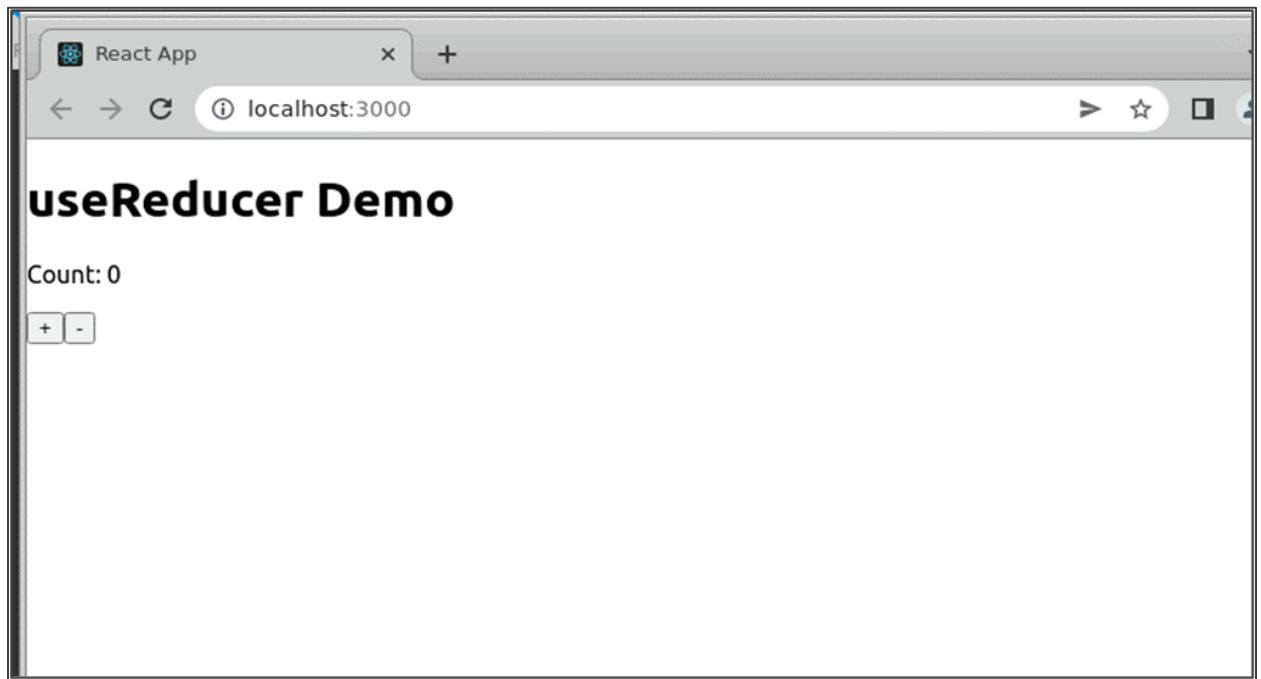
The app should be running, and you should see a simple interface with a counter that can be incremented or decremented by clicking the buttons.

With this, you have successfully created a project to demonstrate the working of **useReducer** Hook.