Ex. No.: 04	How To Move Data From View Controller
Date: 14/08/2024	

Aim:

To create a simple React application that allows users to input their name and two course names, which can then be processed or displayed. This application demonstrates form handling, state management, and user interaction in React.

Procedure:

1. Setup Environment:

- Ensure you have Node.js and npm installed on your system.
- Create a new React application using Create React App:

```
bash

npx create-react-app name-course-collection

cd name-course-collection
```

2. Create the App Component:

- Replace the code in src/App.js with the provided code for collecting user information.

3. Import useState:

- Import useState from React to manage the state for user inputs.

4. Define State Variables:

- Create three state variables using the useState hook:
- name for storing the user's name (initialized to an empty string).
- course1 for storing the first course name (initialized to an empty string).
- course2 for storing the second course name (initialized to an empty string).

5. Handle Form Submission:

- Define a function handleSubmit to handle the form submission:
- Use event.preventDefault() to prevent the default form submission behavior (which would cause a page reload).
- Log the values of name, course1, and course2 to the console.
- Display an alert showing the collected information.
- Clear the input fields by resetting the state variables to empty strings.

6. Render the UI:

- Create a centered layout for the form using inline styles.
- Render a form with three input fields for name and course entries, and a submit button:
- Each input field is linked to its corresponding state variable and updates its value through onChange events.
- The submit button triggers the handleSubmit function when clicked.

7. Styling:

- Use inline styles for the inputs and button to enhance the user interface.

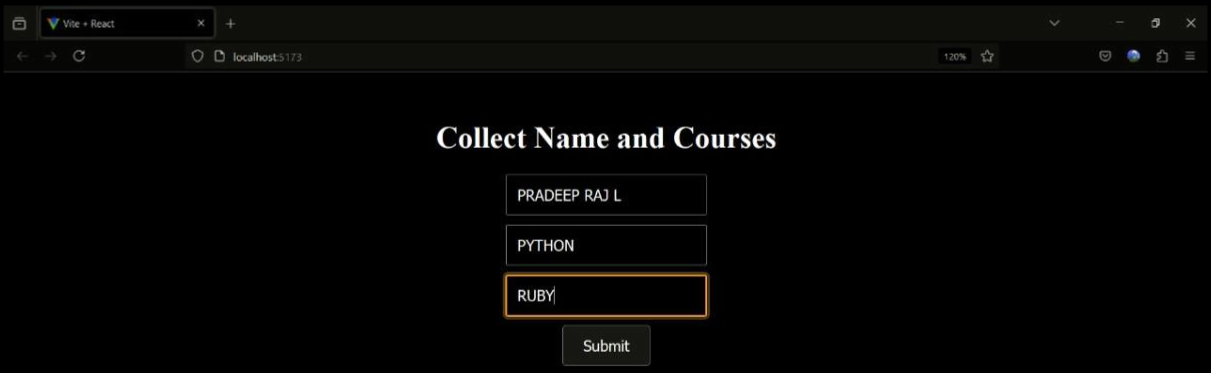
Source Code:

```
import { useState } from 'react';


function App() {
  const [name, setName] = useState(");      // State to store user name
  const [course1, setCourse1] = useState("); // State to store first course
  const [course2, setCourse2] = useState("); // State to store second course


  // Controller function to handle form submission
  const handleSubmit = (event) => {
    event.preventDefault();  // Prevents page reload on form submission


    // Logging the collected data (you can process it further or send it to backend)
    console.log("Name:", name);
    console.log("Course 1:", course1);
    console.log("Course 2:", course2);


    // Example of using the data
    alert(Name: ${name}\nCourse 1: ${course1}\nCourse 2: ${course2});


    // Optionally clear the input fields after submission
    setName(");
    setCourse1(");
    setCourse2(");
  };
```

```
return (
  <div style={{ textAlign: 'center', marginTop: '50px' }}>
    <h1>Collect Name and Courses</h1>

    {/* Form or View */}
    <form onSubmit={handleSubmit}>
      {/* Input for Name */}
      <input
        type="text"
        placeholder="Enter your name"
        value={name}
        onChange={(event) => setName(event.target.value)} // Updates the name state
        style={{ padding: '10px', fontSize: '16px', marginBottom: '10px' }}
      />
      <br />

      {/* Input for Course 1 */}
      <input
        type="text"
        placeholder="Enter course name 1"
        value={course1}
        onChange={(event) => setCourse1(event.target.value)} // Updates the first course state
        style={{ padding: '10px', fontSize: '16px', marginBottom: '10px' }}
      />
      <br />

      {/* Input for Course 2 */}
      <input
        type="text"
        placeholder="Enter course name 2"
        value={course2}
```

```
          onChange={(event) => setCourse2(event.target.value)} // Updates the second course state
          style={{ padding: '10px', fontSize: '16px', marginBottom: '10px' }}
        />
        <br />

        {/* Submit Button */}
        <button type="submit" style={{ padding: '10px 20px', fontSize: '16px' }}>
          Submit
        </button>
      </form>
    </div>
  );
}

export default App;
```

Output:



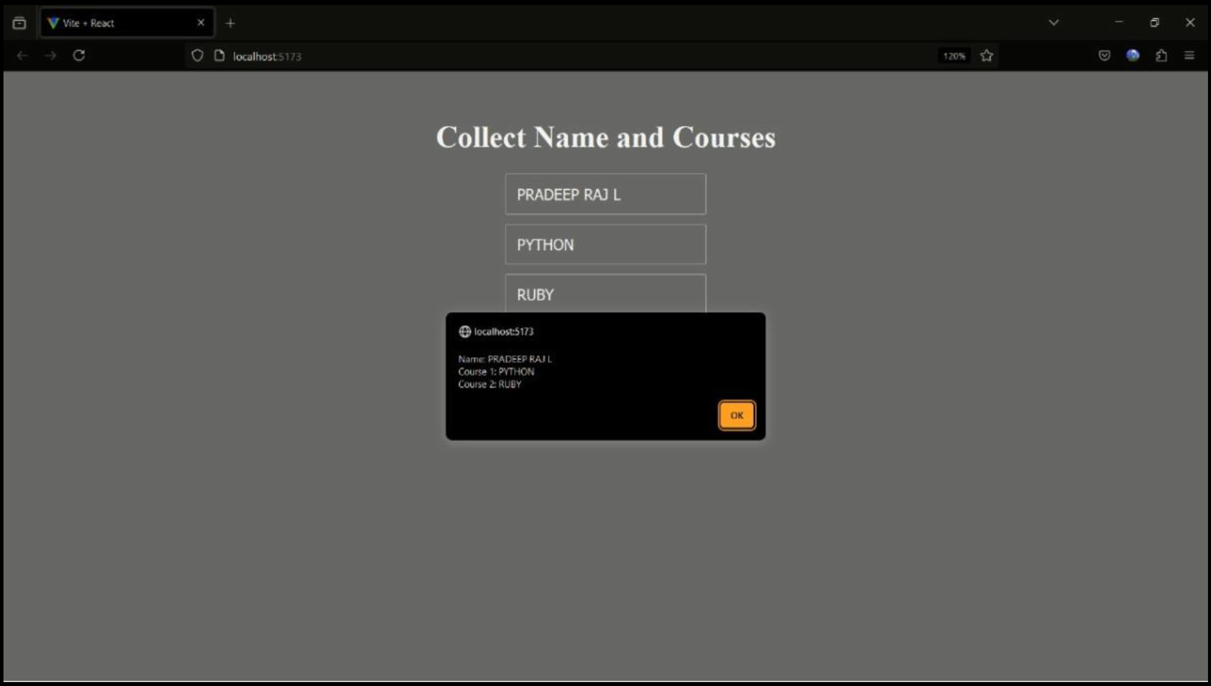
Collect Name and Courses

PRADEEP RAJ L

PYTHON

RUBY

Submit



Collect Name and Courses

PRADEEP RAJ L

PYTHON

RUBY

localhost:5173

Name: PRADEEP RAJ L
Course 1: PYTHON
Course 2: RUBY

Result:

The Name and Courses Collection application allows users to input their name and two course names. Upon submitting the form, the following occurs:

- The application prevents the page from reloading and logs the inputs to the console.
- An alert is displayed showing the collected information:
- Name: Displays the user's name.
- Course 1: Displays the first course name