Experiment 5

Student Name: Nitin   UID: 23BCS12270
Branch: CSE   Section/Group: KRG 3-A
Semester: 6th   Date of Performance:17/02/2026
Subject Name: Full Stack Development – II   Subject Code: 23CSH-309

1. **Aim**: To verify the correctness and reliability of the EcoTrack React application by writing automated tests using Jest and React Testing Library, and by analyzing application behavior using debugging tools.

2. ## Objective:

• Understand the purpose of automated testing in frontend applications
• Write unit tests for JavaScript utility functions using Jest
• Use different Jest matchers to validate expected outputs and behaviors
• Test React components using React Testing Library
• Verify UI rendering by querying elements from the DOM
• Implement asynchronous testing using findBy and waitFor methods
• Apply mocking to simulate API or external data responses in tests
• Perform snapshot testing to detect unintended UI changes
• Debug failing tests and application logic using browser Developer Tools and breakpoints
• Analyze application behavior and errors systematically rather than manual checking

3. ## Implementation / Code:

▪ **Tools & Technologies Used:-**

   • React.js
   • JavaScript (ES6)
   • Jest Testing Framework
   • React Testing Library
   • VS Code
   • Node.js & npm
   • Web Browser (Chrome DevTools)

▪ **Implementation Description:-**

   • The EcoTrack application is tested to ensure correctness of both logic and UI behavior.
   • Unit testing is performed on utility functions (e.g., calculator function) using Jest.
   • React Testing Library is used to render components and verify UI structure.
   • Snapshot testing is applied to detect unintended UI changes over time.

• Automated tests improve application reliability and maintainability.
• Debugging tools such as browser DevTools and breakpoints help identify errors in logic or rendering.

▪ **Sample Code Snippet:-**

```js
// import { render, screen } from "@testing-library/react";
// import Tracker from "./Tracker";

// test("loads async data", async () => {
//    render(<Tracker />);

//    const text = await screen.findByText(/Eco data loaded/i, {}, { timeout: 3000 });

//    expect(text).toBeInTheDocument();
// });

import { render } from "@testing-library/react";
import Tracker from "./Tracker";

test("matches snapshot", () => {
  const { asFragment } = render(<Tracker />);
  expect(asFragment()).toMatchSnapshot();
});
```

```js
import { add } from "./calc";

test("adds two numbers", () => {
    expect(add(2, 3)).toBe(5);
});
```