Student Management System

Let's build a website. Instead of just showing static pages, it will be **dynamic and interactive**. This project uses **React**, a special toolbox for building websites with reusable parts.

1. The Building Blocks: Components

Think of your website like a LEGO model. Each LEGO brick is a **component**.

- **Navbar** → The top navigation bar with a logo and links, reused on every page.
- **StudentCard** → A small block showing a student's details (name, email, course).
- **StudentForm** → A form for adding or editing a student, with inputs and save logic.

Components make the code **clean, organized, and reusable**.

2. Remembering Things: State

Dynamic websites need memory. This memory is called **state**.

- name stores the student's name.
- setName updates it.

When state changes, React **automatically updates the screen**.

3. Passing Information: Props

Components talk to each other using **props** (short for properties).

- The student data is passed down.
- StudentCard then shows the correct details.

Props make components **flexible and reusable**.

4. Working Behind the Scenes: Hooks

Hooks are special functions that give components superpowers.

- **useEffect** → Runs code after a component loads (fetching courses from a server).
- **useContext** → Shares data (like the student list) across many components without passing props manually.

5. Moving Between Pages: Routing

Websites often have multiple pages. React uses **React Router** for smooth navigation.

- /students → Student list page
- /add → Add student page

React Router swaps components instead of reloading the whole site → **fast and smooth experience**.

Summary:

This project shows how modern React apps work:

- **Components** → Reusable building blocks
- **State** → Memory of a component
- **Props** → How components share info
- **Hooks** → Extra powers like fetching data and sharing context
- **Routing** → Navigation without page reloads

By combining these, we can build **powerful, efficient, and user-friendly web apps**.

Flow Diagram :