

AlertContext.js

```
import { createContext, useReducer } from "react";
import alertReducer from "../AlertReducer";

const AlertContext = createContext();

export const AlertProvider = ({ children }) => {
  const initialState = null; // we set null as default, we'll
  // manipulate this start below.

  const [state, dispatch] = useReducer(alertReducer, initialState);

  // Set an Alert

  const setAlert = (msg, type) => {
    dispatch({
      type: "SET_ALERT", // we set a type here, we'll use this as
      // case at AlertReducer
      payload: { msg, type }
    })

    setTimeout(() => dispatch({ type: "REMOVE_ALERT" }), 3000)
  }

  return <AlertContext.Provider value={{ hamza: state, setAlert }}>
    {children}
  </AlertContext.Provider>
}

export default AlertContext
```

AlertReducer.js

```
const alertReducer = (state, action) => {
  switch (action.type) {
    case "SET_ALERT":
      return action.payload
```

```

    case "REMOVE_ALERT": / When it happens 3secs later, it won't
show.
    return null
  default:
    return state
  }
}

export default alertReducer

```

```

import { useContext } from "react";
import AlertContext from "../../context/alert/AlertContext";

function Alert() {
  const { alert } = useContext(AlertContext);

  return (
    alert !== null && ( /
    <div className="flex items-start mb-4 space-x-2">
      {alert.type === "error" && (
        <svg
          className="w-6 h-6 flex-none mt-0.5"
          fill="none"
          viewBox="0 0 24 24"
        >
          <g>
            <path d="M8 8 16 8" stroke="#B91C1C"
strokewidth="2" />
          </g>
        </svg>
      )}
      <p className="flex-1 text-base font-semibold leading-7
text-white">
        <strong>{alert.msg}</strong>

```

```

        </p>
      </div>
    )
  );
}

export default Alert;

```

```

import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
import Home from "../pages/Home";
import NotFound from "../pages/NotFound";
import About from "../pages/About";
import Navbar from "../components/layout/Navbar";
import Footer from "../components/layout/Footer";
import Alert from "../components/layout/Alert";
import { GithubProvider } from "../context/github/GithubContext";
import { AlertProvider } from "../context/alert/AlertContext";

function App() {
  return (
    <GithubProvider>
      <AlertProvider> {/ We wrap everything AlertProvider like GithubProvider
        <Router>
          <div className="flex flex-col justify-between h-screen">
            <Navbar />
            <main className="container mx-auto px-3 pb-12">
              <Alert /> {/ We add Alert Component here.
                <Routes>
                  <Route path="/" element={<Home />} />
                  <Route path="/about" element={<About />} />
                  <Route path="/notfound" element={<NotFound />} />
                  <Route path="/*" element={<NotFound />} />
                </Routes>
              </main>
            </div>
          </Router>
        </AlertProvider>
      </GithubProvider>
    )
  }
}

```

```

        </main>
        <Footer />
      </div>
    </Router>
  </AlertProvider>
</GithubProvider>
);
}

export default App;

```

UserSearch.jsx

```

import { useState, useContext } from "react";
import GithubContext from "../../context/github/GithubContext";
import AlertContext from "../../context/alert/AlertContext";

function UserSearch() {
  const [text, setText] = useState("");

  const { users, searchUsers, clearUsers } =
    useContext(GithubContext);

  const { setAlert } = useContext(AlertContext); /

  const handleChange = (e) => setText(e.target.value);
  const handleSubmit = (e) => {
    e.preventDefault();

    if (text === "") { / We use setAlert here, because our input is
here. After that when we click go button without text, our state
will change as below.(Because we used reducer.)
      setAlert("Please Enter Something", "error");
    } else {
      searchUsers(text);
      setText("");
    }
  }
}

```

```

    }
  };

  return (
    <div className="grid grid-cols-1 xl:grid-cols-2 lg:grid-cols-2
md:grid-cols-2 mb-8 gap-8">
      <div>
        <form onSubmit={handleSubmit}>
          <div className="form-control">
            <div className="relative">
              <input
                type="text"
                className="w-full pr-40 bg-gray-200 input input-lg
text-black"
                placeholder="Search"
                value={text}
                onChange={handleChange}
              />
              <button
                type="submit"
                className="absolute top-0 right-0 rounded-l-none w-
36 btn btn-lg"
              >
                Go
              </button>
            </div>
          </div>
        </form>
      </div>
      {users.length > 0 && (
        <div>
          <button onClick={clearUsers} className="btn btn-ghost
btn-lg">
            Clear
          </button>
        </div>
      )}
    </div>
  );
}

export default UserSearch;

```

