Name: Likhith B A
Reg. No: 23BCE1098

Experiment No. 15
React JSX

Code:

JSX COMPONENTS:

```
import React, {
  useState,
  useReducer,
  useEffect,
  useRef,
  createContext,
  useContext,
} from "react";
import PropTypes from "prop-types";
const Header = ({ title }) => <h1>{title}</h1>;
const Content = () => {
  const [joke, setJoke] = useState("");
  const fetchJoke = async () => {
    const res = await fetch("https://api.chucknorris.io/jokes/random");
    const data = await res.json();
    setJoke(data.value);
  };
  useEffect(() => {
    fetchJoke();
  }, []); // Fetch joke only once on mount
  return (
    <div>
      <button onClick={fetchJoke}>Get Joke</button>
      <p>{joke}</p>
    </div>
  );
};

// 3. Footer Component
const Footer = () => <p>Static Footer Message</p>;
```

```
// 4. Styled Button
const StyledButton = () => (
  <button
    style={{
      backgroundColor: "blue",
      padding: "10px",
      fontSize: "16px",
      color: "white",
    }}
  >
    Click Me
  </button>
);
class LifecycleDemo extends React.Component {
  componentDidMount() {
    console.log("Mounted");
  }
  componentDidUpdate() {
    console.log("Updated");
  }
  componentWillUnmount() {
    console.log("Unmounted");
  }
  render() {
    return <div>Lifecycle Component</div>;
  }
}
const Counter = () => {
  const [count, dispatch] = useReducer((state, action) => {
    switch (action.type) {
      case "INC":
        return state + 1;
      case "DEC":
        return state - 1;
      default:
        return state;
    }
  }, 0);
  return (
    <div>
```

```
      <p>Count: {count}</p>
      <button onClick={() => dispatch({ type: "INC" })}>+</button>
      <button onClick={() => dispatch({ type: "DEC" })}>-</button>
    </div>
  );
};
const JokeComponent = () => {
  const [joke, setJoke] = useState("");
  const fetchJoke = async () => {
    const res = await fetch("https://api.chucknorris.io/jokes/random");
    const data = await res.json();
    setJoke(data.value);
  };
  useEffect(() => {
    fetchJoke();
  }, []); // Fetch joke only once on mount
  return (
    <div>
      <p>{joke}</p>
      <button onClick={fetchJoke}>Refresh</button>
    </div>
  );
};
const FocusForm = () => {
  const inputRef = useRef();
  return (
    <div>
      <input ref={inputRef} />
      <button onClick={() => inputRef.current.focus()}>Focus</button>
    </div>
  );
};
const ThemeContext = createContext();
const ThemeSwitcher = () => {
  const { theme, toggleTheme } = useContext(ThemeContext);
  return (
    <button
      onClick={toggleTheme}
      style={{ background: theme === "dark" ? "#333" : "#FFF" }}
    >
```

```
      Toggle Theme ({theme})
    </button>
  );
};
const Parent = () => <Child message="Hello from Parent!" />;
const Child = ({ message }) => <p>{message}</p>;
Child.propTypes = { message: PropTypes.string.isRequired };

// 11. Forms
const FormState = () => {
  const [values, setValues] = useState({ name: "", email: "" });
  return (
    <form onSubmit={(e) => e.preventDefault()}>
      <input
        value={values.name}
        onChange={(e) => setValues((v) => ({ ...v, name: e.target.value }))}
      />
      <input
        value={values.email}
        onChange={(e) => setValues((v) => ({ ...v, email: e.target.value }))}
      />
      <p>
        Live: {values.name} - {values.email}
      </p>
    </form>
  );
};
const FormRef = () => {
  const nameRef = useRef();
  const emailRef = useRef();
  const handleSubmit = (e) => {
    e.preventDefault();
    alert(`${nameRef.current.value} - ${emailRef.current.value}`);
  };
  return (
    <form onSubmit={handleSubmit}>
      <input ref={nameRef} />
      <input ref={emailRef} />
      <button>Submit</button>
    </form>
  );
};
const Components = () => {
  const [theme, setTheme] = useState("light");
  const toggleTheme = () => setTheme(theme === "dark" ? "light" : "dark");
  return (
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
      <div>
        <Header title="React Components Demo" />
        <Content />
        <Footer />
        <StyledButton />
        <LifecycleDemo />
        <Counter />
        <JokeComponent />
        <FocusForm />
        <ThemeSwitcher />
        <Parent />
        <FormState />
        <FormRef />
      </div>
    </ThemeContext.Provider>
  );
};
export default Components;
```

App.JSX

```
import React from "react";
import "./app.css"
import Components from "./Components";

function App() {
  return (
    <div className="App">
      <Components />
    </div>
  );
}
```

```
export default App;
```

StyleSheets

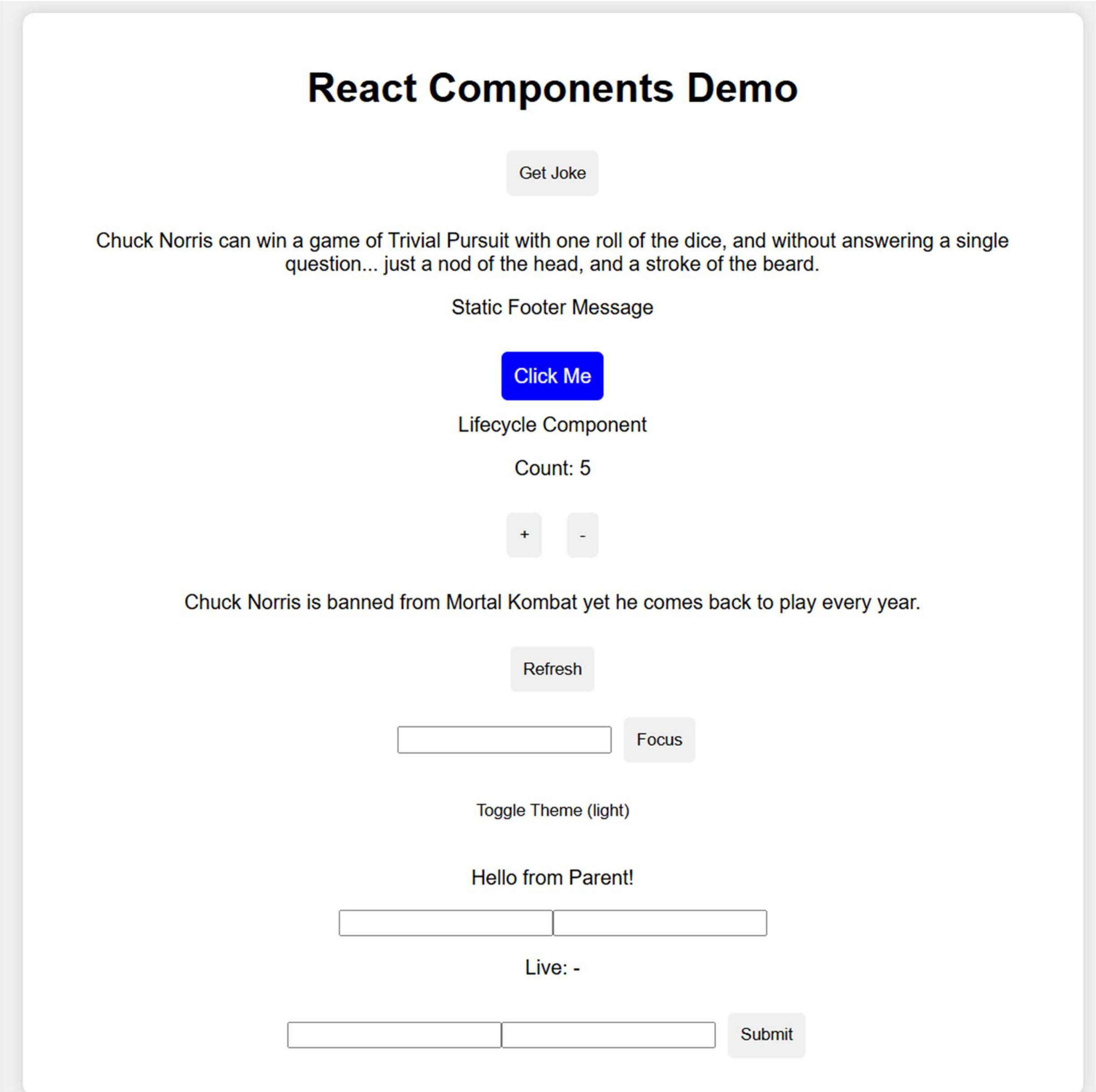
```
body {
  font-family: Arial, sans-serif;
  padding: 20px;
  margin: 0;
  background-color: #f0f0f0;
}

.App {
  text-align: center;
  max-width: 800px;
  margin: auto;
  padding: 20px;
  background: #fff;
  border: 1px solid #ddd;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

button {
  padding: 10px;
  margin: 10px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #ccc;
}
```

OUTPUT :



React Components Demo

Chuck Norris can win a game of Trivial Pursuit with one roll of the dice, and without answering a single question... just a nod of the head, and a stroke of the beard.

Static Footer Message

Lifecycle Component

Count: 5

Chuck Norris is banned from Mortal Kombat yet he comes back to play every year.

Hello from Parent!

Live: -