Learner Assignment Submission Format

Learner Details
• Name: Nikil G S
• Enrollment Number:
• Batch / Class: MERN STACK
• Assignment: Toggle Theme using React
• Date of Submission: 11/08/2025

Problem Solving Activity 1.1

1. Program Statement

The program allows the user to toggle between Light and Dark themes in a React application using the Context API. The Navbar component displays the current theme and a button to switch themes.

---

2. Algorithm

1. Create a ThemeContext using createContext() to hold the theme data and toggle function.
2. Create a ThemeProvider component:
    - ○ Initialize theme state as "light".
    - ○ Create toggleTheme function to switch between "light" and "dark".

---

3. Pseudocode

```
START


CREATE ThemeContext


FUNCTION ThemeProvider(children):
    theme ← "light"
    FUNCTION toggleTheme():
        IF theme == "light" THEN
```

```
      theme ← "dark"
    ELSE
      theme ← "light"
  RETURN ThemeContext.Provider({theme, toggleTheme}) with children

FUNCTION Navbar():
  GET theme, toggleTheme FROM ThemeContext
  SET style based on theme
  DISPLAY "Current Theme: theme"
  DISPLAY button "Toggle Theme" ON click → toggleTheme

MAIN APP:
  WRAP Navbar inside ThemeProvider

END
```

---

4. Program Code

ThemeProvider.jsx

```
import React, { createContext, useState } from 'react';

export const ThemeContext = createContext();

const ThemeProvider = ({ children }) => {
  const [theme, setTheme] = useState("light");

  const toggleTheme = () => {
    setTheme(prevTheme => (prevTheme === "light" ? "dark" : "light"));
  };

  return (
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
      {children}
    </ThemeContext.Provider>
  );
};

export default ThemeProvider;
```

Navbar.jsx

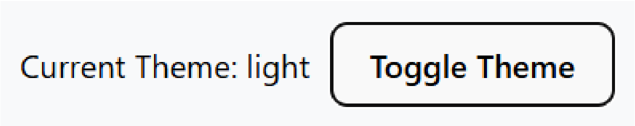
```
import React, { useContext } from 'react';
import { ThemeContext } from './ThemeProvider';

const Navbar = () => {
  const { theme, toggleTheme } = useContext(ThemeContext);

  const navbarStyle = {
    backgroundColor: theme === "light" ? "#f8f9fa" : "#343a40",
    color: theme === "light" ? "#000" : "#fff",
    padding: "10px",
    textAlign: "center",
  };

  return (
    <nav style={navbarStyle}>
      Current Theme: {theme}
      <button
        onClick={toggleTheme}
        style={{ marginLeft: '10px', marginTop: '5px' }}
      >
        Toggle Theme
      </button>
    </nav>
  );
};

export default Navbar;
```

---

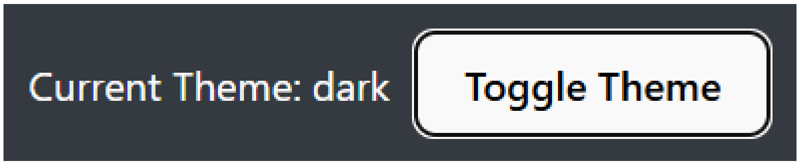
5. Test Cases

Test Case	Input	Expected Output	Result
1	Initial load	Light theme with light background & black text	Pass
2	Click toggle	Dark theme with dark background & white text	Pass
3	Click toggle again	Switch back to light theme	Pass

Out put.



Current Theme: light



Current Theme: dark

6. Observation / Reflection

While building this project, I learned how to use the React Context API to share state between components without prop drilling. It also helped me understand dynamic styling based on state values. For future improvements, I plan to:

Problem Solving Activity 1.2

Follow the same Structure as problem Solving Activity 1.1.