

## Assignment

### Learner Details

- **Name:** Sariya Mazhar
  - **Enrollment Number:** SU625MR002
  - **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 a user to switch between **Light** and **Dark** themes in a React application.

- **Input:** A button click to toggle the theme.
- **Output:** The application's background and text colors change based on the current theme (Light or Dark).

The program uses **React Context API** to share the theme state across multiple components (NavBar and ThemeToggle) without passing props manually.

---

#### 2. Algorithm

- Create a ThemeContext using createContext() to store theme data and a toggle function.
- Create a ThemeProvider component to hold:
  - theme state (light or dark)
  - toggleTheme function to switch between themes.
- Wrap application components inside the ThemeProvider to share theme data.
- In NavBar component:
  - Access the current theme from ThemeContext.
  - Apply different background and text colors based on theme value.
- In ThemeToggle component:
  - Access the toggleTheme function from ThemeContext.

- On button click, call toggleTheme to switch themes.
  - Then we render both components and verify that the theme changes dynamically.
- 

### 3. Pseudocode

START

CREATE ThemeContext

FUNCTION ThemeProvider(children):

    SET theme = "dark"

    FUNCTION toggleTheme():

        IF theme == "light" THEN

            theme = "dark"

        ELSE

            theme = "light"

    RETURN ThemeContext.Provider(value={theme, toggleTheme}) containing children

FUNCTION NavBar():

    GET theme from ThemeContext

    SET navbarStyle:

        background = light? "#f0f0f0" : "#333"

        color = light? "#000" : "#fff"

    DISPLAY nav bar with current theme name

FUNCTION ThemeToggle():

    GET toggleTheme from ThemeContext

    DISPLAY button "Toggle Theme"

    ON button click CALL toggleTheme

MAIN APP:

    WRAP NavBar and ThemeToggle inside ThemeProvider

END

---

#### 4. Program Code

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

const Navbar = () => {
    const {theme}=useContext(ThemeContext)

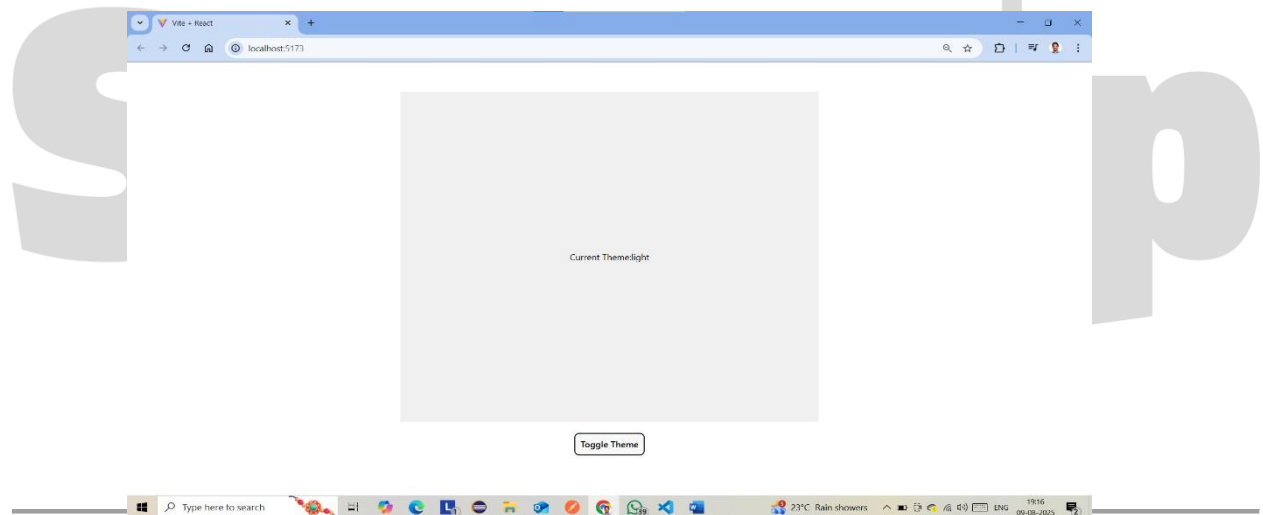
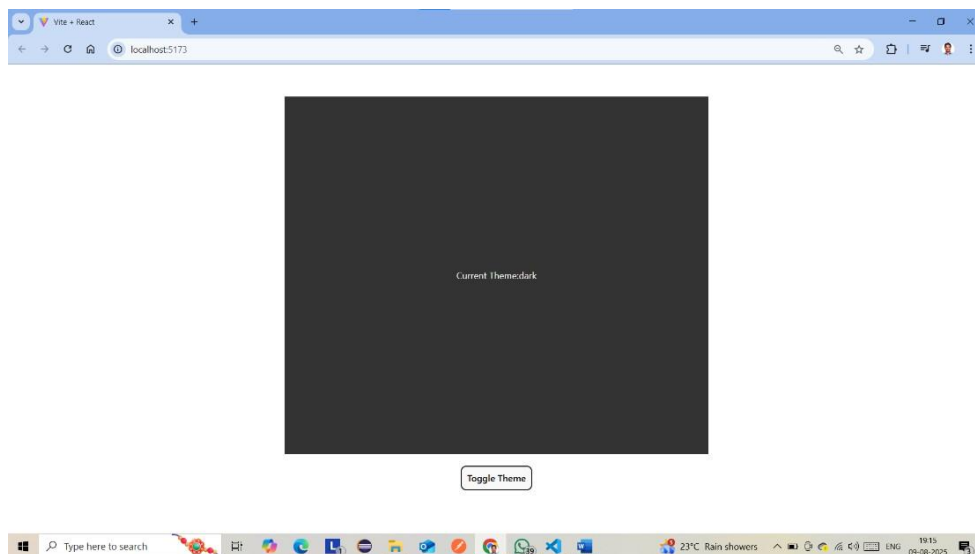
    const navbarStyle={
        padding:"50px",
        backgroundColor:theme=="light"?"#f9f8f8ff":"#333",
        color:theme=="light"?"#000":"#fff"
    }

    return (
        <div>
            <nav style={navbarStyle}>Current Theme: {theme}</nav>
        </div>
    )
}

export default Navbar
```

---

## 5. Screenshots of Output



## 7. Observation / Reflection

Implementing a toggle mode in React using `useState` (or `useContext`), combined with `useEffect` and `localStorage`, provides a seamless, persistent theme switch—enhanced by using CSS variables for clean, scalable styling.