

## Learner Details

- **Name:** Nikhil k
  - **Enrollment Number:** su625mr004
  - **Batch / Class:** Mern Stack
  - **Assignment:** Theme toggle
  - **Date of Submission:** 11-08-2025
- 

## Problem Solving Activity 1.1

### 1. Program Statement

Create a React application that uses the Context API to manage and toggle between light and dark themes. The selected theme should be applied to the navbar, and a button should allow switching themes globally.

---

### 2. Algorithm

- Create a ThemeContext using createContext().
  - In the Toggle component, store the theme state (light by default) using useState.
  - Define a function toggleTheme() to switch between "light" and "dark".
  - Provide theme and toggleTheme values to child components through ThemeContext.Provider.
  - In the Navbar component, use useContext(ThemeContext) to consume these values.
  - Apply theme to navbar styles dynamically and use the toggle button to change themes.
- 

### 3. Pseudocode

START

CREATE ThemeContext

INITIALIZE theme as "light"

DEFINE toggleTheme to switch between light and dark

WRAP children components with ThemeContext.Provider

IN Navbar:

USE useContext to access theme and toggleTheme

SET navbar style based on theme value

ON button click, call toggleTheme

END

---

#### 4. Program Code

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

```
export const ThemeContext=createContext();
```

```
export const Toggle = ({children}) => {
```

```
  const[theme,setTheme]=useState("light");
```

```
  const toggleTheme=()=>{
```

```
    setTheme((prevTheme)=>(prevTheme=="light"? "dark": "light"));
```

```
  }
```

```
  return (
```

```
    <div>
```

```
      <ThemeContext.Provider value={{theme,toggleTheme}}>
```

```
        {children}
```

```
      </ThemeContext.Provider>
```

```
    </div>
```

```
  )
```

```
}
```



Stemup  
A Unit of Pragnova Pvt Ltd

```
export default Toggle
```

```
import React,{useContext} from 'react'
```

```
import { ThemeContext } from './Toggle'
```

```
const Navbar = () => {
```

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

```
  const navbarStyle={
```

```
    padding:"10px",
```

```
    backgroundColor:theme==="light"?"#f0f0f0":"#333",
```

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

```
  }
```

```
  return (
```

```
    <div>
```

```
      <nav style={navbarStyle}>Current Theme: {theme}</nav>
```

```
      <button onClick={toggleTheme}>THEME</button>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default Navbar
```

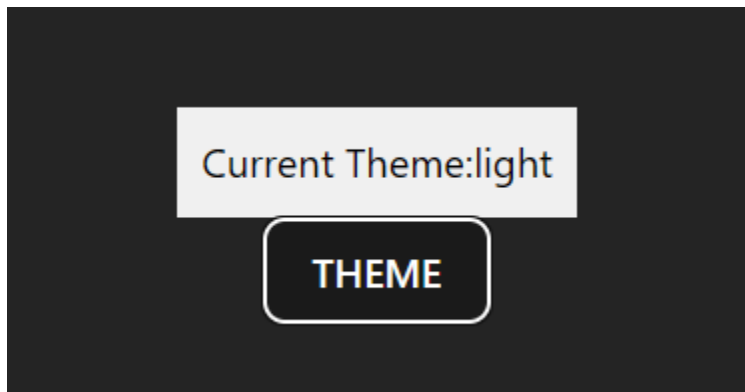
---

## 5. Test Cases

Initial Theme	Action	Expected Theme
light	Click "THEME" button	dark
dark	Click "THEME" button	light
light	Multiple clicks	Alternates each time

---

## 6. Screenshots of Output



---

## 7. Observation / Reflection

- Learned how to use the Context API to share state between components without prop drilling.
- Understood how to style components dynamically based on state values.
- The toggle button demonstrates state updates across the entire app instantly.