Learner Assignment Submission Format

Learner Details

- **Name:H O Akash**
- **Enrollment Number:SU625MR015**
- **Batch / Class:MERN Stack**
- **Assignment:Mini Project**
- **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 a theme (light or dark) across multiple components. The application should include a navbar that displays the current theme and a button to toggle between light and dark themes.

---

2. Algorithm

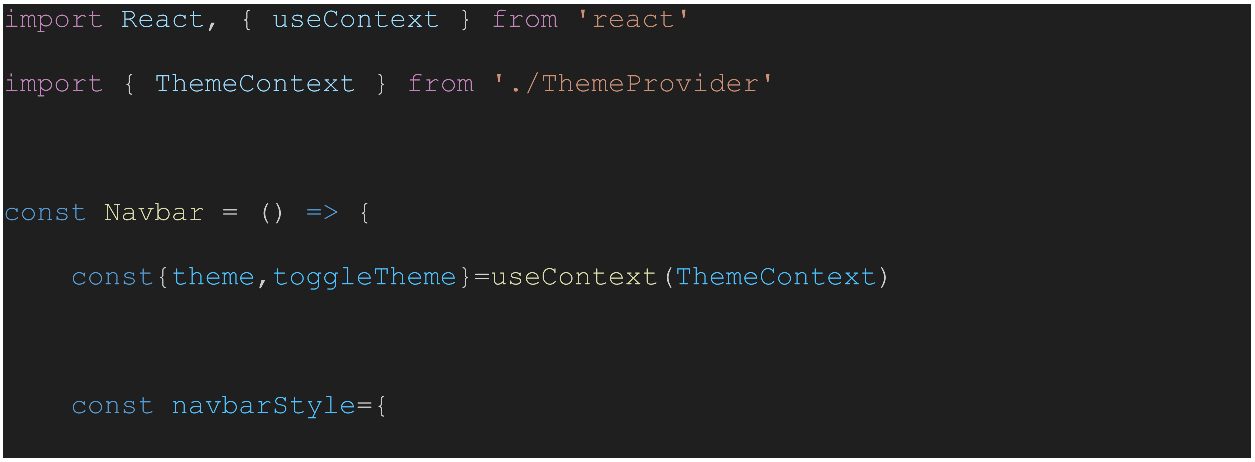
```
BEGIN
 DEFINE ThemeContext
 CREATE ThemeProvider component
   INITIALIZE theme state with "light" value
   DEFINE toggleTheme function to switch between light and dark themes
   PROVIDE theme and toggleTheme to child components
 DEFINE Navbar component
   USE useContext hook to access theme and toggleTheme
   RENDER navbar with current theme and toggle button
END
```

---

3. Pseudocode

```
BEGIN
  DEFINE ThemeContext using createContext
  CREATE ThemeProvider component
    INITIALIZE theme state with "light" value using useState
    DEFINE toggleTheme function
      IF theme is "light"
        SET theme to "dark"
      ELSE
        SET theme to "light"
    PROVIDE theme and toggleTheme to child components using ThemeContext.Provider
  DEFINE Navbar component
    USE useContext hook to access theme and toggleTheme from ThemeContext
    DEFINE navbar style based on current theme
    RENDER navbar with current theme and toggle button
    ATTACH toggleTheme function to button click event
END
```

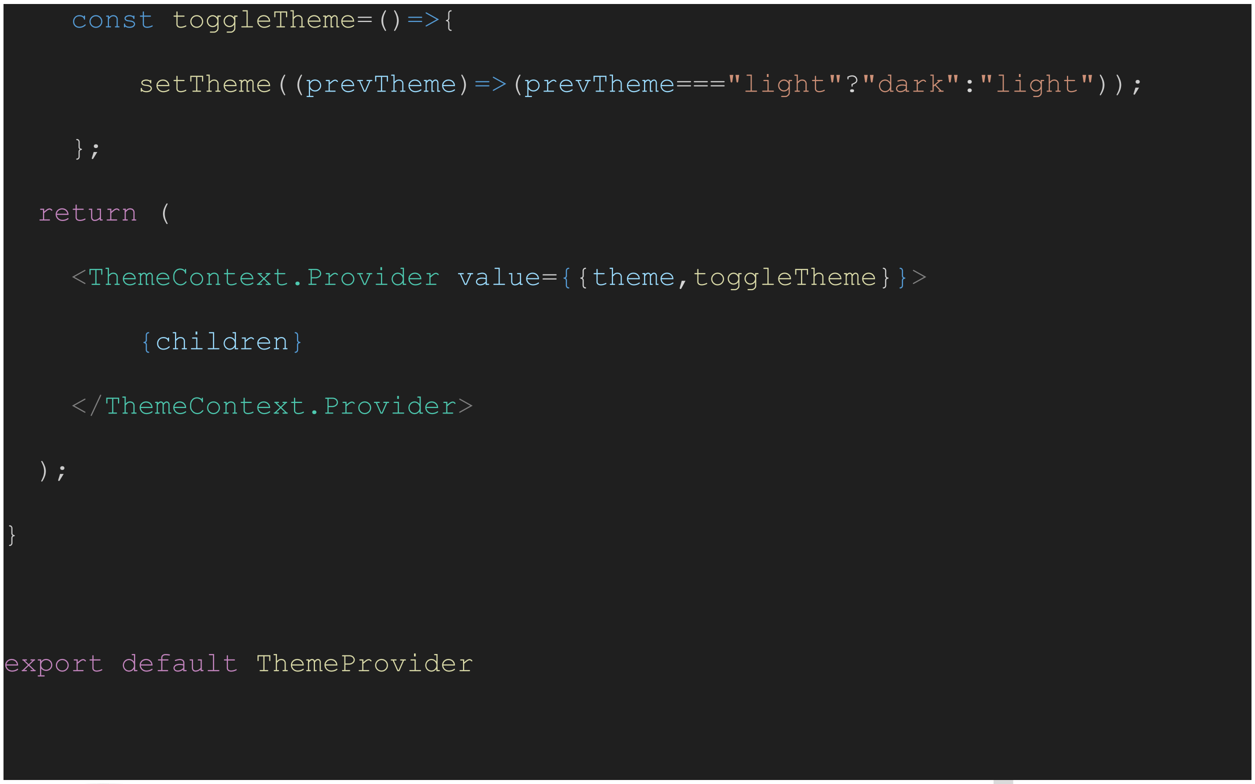
4. Program Code(Navbar.jsx)



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

import { ThemeContext } from './ThemeProvider'


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><br /><br />

     <button onClick={toggleTheme} style={{
backgroundColor:theme==="light"?"#f0f0f0":"#333",

        color:theme==="light"?"#000":"#fff"}}>{theme}</button>

</div>

  )

};


export default Navbar
```

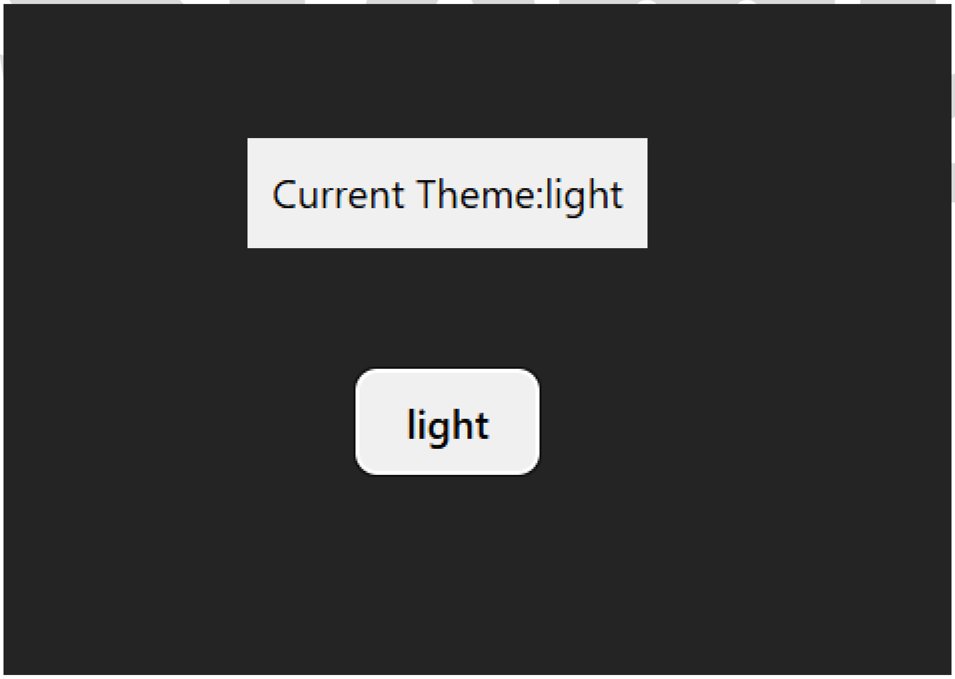
5. Program Code(ThemeProvider.jsx)

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

export const ThemeContext=createContext();

// import { Children } from 'react';


export const ThemeProvider = ({children}) => {

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

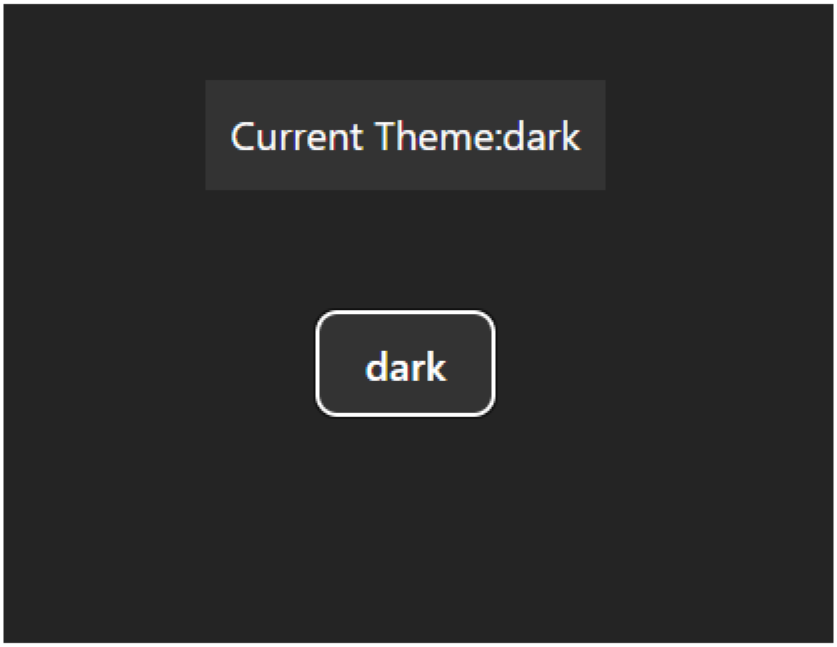


```
    const toggleTheme=()=>{
        setTheme((prevTheme)=>(prevTheme==="light"?"dark":"light"));
    };
  return (
    <ThemeContext.Provider value={{theme,toggleTheme}}>
        {children}
    </ThemeContext.Provider>
  );
}

export default ThemeProvider
```

6. Screenshots of Output



Current Theme:light



Current Theme:dark

7. Observation / Reflection

The given React code is a simple implementation of theme management using the Context API. Here are some reflections:

- The code uses the createContext function to create a ThemeContext that can be shared across multiple components.

- The ThemeProvider component wraps the application and provides the theme state and toggleTheme function to child components.

- The Navbar component uses the useContext hook to access the theme and toggleTheme function from the ThemeContext.

- The code uses a simple state management approach to toggle between light and dark themes.

Problem Solving Activity 1.2

Follow the same Structure as problem Solving Activity 1.1.