**Steps to Set Up React Context**

**1. Create Context**

jsx

Copy code

// src/contexts/ThemeContext.js

import React from 'react';

const ThemeContext = React.createContext();

export default ThemeContext;

**2. Create a Provider Component**

jsx

Copy code

// src/contexts/ThemeProvider.js

import React, { useState } from 'react';

import ThemeContext from './ThemeContext';

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;

**3. Use Context in Components**

jsx

Copy code

// src/components/ThemeSwitcher.js

import React, { useContext } from 'react';

import ThemeContext from '../contexts/ThemeContext';

const ThemeSwitcher = () => {

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

return (

<div>

<p>Current Theme: {theme}</p>

<button onClick={toggleTheme}>Toggle Theme</button>

</div>

);

};

export default ThemeSwitcher;

**4. Integrate Provider in App**

jsx

Copy code

// src/App.js

import React from 'react';

import ThemeProvider from './contexts/ThemeProvider';

import ThemeSwitcher from './components/ThemeSwitcher';

const App = () => {

return (

<ThemeProvider>

<div className="App">

<ThemeSwitcher />

</div>

</ThemeProvider>

);

};

export default App;