### React Hook – Context & Effect Hook

#### Objectives and Outcomes

In this exercise we understand about how to using React Hook: Context & Effect Hook . At the end of this exercise you will learn about:

* Using the React Hook such as: Context & Effect Hook in your app.

#### Add Context & Effect Hook to your application for toggle the dark theme to the light theme at navigation bar

* Create the ThemeContext.js file in components folder, add some code as follows:

const themes = {

  dark: {

    backgroundColor: 'black',

    color: 'white'

  },

  light: {

    backgroundColor: 'white',

    color: 'black'

  }

}

const initialState = {

  dark: false,

  theme: themes.light,

  toggle: () => {}

}

* React's Context is initialized as follow:

const ThemeContext = React.createContext(initialState)

* Create a method which wraps all children with ThemeContext.Provider component and export this method and the actual ThemeContext object that we created just before.

function ThemeProvider({ children }) {

  const [dark, setDark] = useState(false) // Default theme is light

  // On mount, read the preferred theme from the persistence

  useEffect(() => {

    const isDark = localStorage.getItem('dark') === 'true'

    setDark(isDark)

  }, [dark])

  // To toggle between dark and light modes

  const toggle = () => {

    const isDark = !dark

    localStorage.setItem('dark', JSON.stringify(isDark))

    setDark(isDark)

  }

  const theme = dark ? themes.dark : themes.light

  return (

    <ThemeContext.Provider value={{ theme, dark, toggle }}>

      {children}

    </ThemeContext.Provider>

  )

}

export { ThemeProvider, ThemeContext }

* **Remember to import the Effect and State Hook**

import { useState,  useEffect } from 'react'

* The final ThemeContext.js file looks like:

import React from 'react'

import { useState,  useEffect } from 'react'

const themes = {

  dark: {

    backgroundColor: 'black',

    color: 'white'

  },

  light: {

    backgroundColor: 'white',

    color: 'black'

  }

}

const initialState = {

  dark: false,

  theme: themes.light,

  toggle: () => {}

}

const ThemeContext = React.createContext(initialState)

function ThemeProvider({ children }) {

  const [dark, setDark] = useState(false)

  // On mount, read the preferred theme from the persistence

  useEffect(() => {

    const isDark = localStorage.getItem('dark') === 'true'

//store the state mode to the local storage

    setDark(isDark)

  }, [dark])

  // To toggle between dark and light modes

  const toggle = () => {

    const isDark = !dark

    localStorage.setItem('dark', JSON.stringify(isDark))

    setDark(isDark)

  }

  const theme = dark ? themes.dark : themes.light

  return (

    <ThemeContext.Provider value={{ theme, dark, toggle }}>

      {children}

    </ThemeContext.Provider>

  )

}

export { ThemeProvider, ThemeContext }

* Wrap all our application in ThemeContext.Provider at index.js file

…

import { ThemeProvider } from './components/ThemeContext';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

     <ThemeProvider>

    <App />

    </ThemeProvider>

  </React.StrictMode>

);

* Update the Navigation.js component to perform the switch theme function as follow:

import React from 'react'

import { useContext } from 'react'

import { ThemeContext } from './ThemeContext'

export default function Navigation() {

  const { theme, toggle, dark } = useContext(ThemeContext)

  return (

    <div>

      <nav style={{ backgroundColor: theme.backgroundColor, color: theme.color }}>

          <ul>

              <li><a className='active' href='#home'>Home</a></li>

              <li><a href='#news'>News</a></li>

              <li><a href='#about'>About</a></li>

              <li><a href='#contact'>Contact</a></li>

</ul>

          <div style={{position: 'relative'}}>

          <a className='switch-mode' href='#' onClick={toggle}

          style={{

            backgroundColor: theme.backgroundColor,

            color: theme.color,

            outline: 'none'

          }} data-testid="toggle-theme-btn"

        >

          Switch Nav to {!dark ? 'Dark' : 'Light'} mode

         </a>

         </div>

      </nav>

    </div>

  )

}

#### Screen display in Dark navigation bar

#### 

#### Screen display in Light navigation bar A collage of a person Description automatically generated with low confidence

#### Conclusions

In this exercise you learnt how to use Effect & Context Hook in your app.