Side-effects:

When ever state updates it re-render whole component. So when http request(sideeffect) was their in also run again. So we use useeffect() hooks

Use effect:

UseEffect run after all component renders and and when dependency changes

If dependency is not changed it run only once.

  useEffect(()=>{

    const storedUserLoggedInformation = localStorage.getItem('isLoggedIn')

    if(storedUserLoggedInformation =='1'){

      setIsLoggedIn(true);

    }

  },[])

In this example ui renders first..then use effect run ..the in setIsLoggedIn state changed then ui re renders

Without dependency:

  useEffect(()=>{

    setFormIsValid(

      enteredEmail.target.value.includes('@') && enteredPassword.trim().length > 6

    );

  })

As same as without use effect..run every time when ui rerenders

With dependencies:

  useEffect(()=>{

    setFormIsValid(

      enteredEmail.target.value.includes('@') && enteredPassword.trim().length > 6

    );

  },[enteredEmail,enteredPassword])