

use effect hook

- The syntax for use effect hook:

```
useEffect(()=>{}, [dependencies]);
```

- The function should be executed after every component evaluation if the specified dependencies change.
- Dependencies of this effect the function only runs if the dependencies change
- .Small example to understand useEffect():

```
import React, {useState, useEffect} from 'react';
function App={()=>
{
const [isLoggedIn, setIsLoggedIn]=useState(false);
useEffect(()=>{
const storedUserLoggedInInformation=localStorage.getItem("login");
if(storedUserLoggedInInformation==="1")
{
setLoggedIn(true);
}
}, []);
const loginHandler={()=>
{
localStorage.setItem("login", "1");//this is stored in local storage
setIsLoggedIn(true);
}
const loginHandler={()=>
{
localStorage.removeItem("login");
setIsLoggedIn(false);
}
return(
<div>
<button onClick={loginHandler}>login</button>
<button onClick={logoutHandler}>logout</button>
</div>
);
}
```

```
import React, {useEffect} from 'react';

function Login={()=>
```

```

{
  const [formIsValid,setFormIsValid]=useState(false);
  useEffect(()=>{
    const identifier=setTimeout(()=>{
      setFormIsValid(enteredEmail.include('@') && enteredPassword>6);
    },500);
    return ()=>{
      clearTimeout('identifier');
    };
  },[enteredEmail,enteredPassword]);
  const formHandler=(()=>
  {
    setFormIsValid(true);
  }
  )
  return(
    <div onclick={formHandler}>login</button>
  )
}

```