National University of Computer and Emerging Sciences



Web Programming

Assignment 7

|  |  |
| --- | --- |
| Roll Number | 19F-0228 |
| Name | Muhammad Zain |
| INSTRUCTOR | Sir Shehroze khan |
| Semester | FALL 2021 |

Hooks

# 1: Use state Hook:

# Description:

Usestate is used to set the initial value and update the state.

# Code Snippet:

const onDelete =(*todo*)=>{

console.log("I am onDelete",*todo*);

setTodos(todos.filter((*tasks*)=>{

return *tasks*!==*todo*;

}))

}

const [todos,setTodos]= useState([

{ sno:1,

title:"Go to the market",

desc: "Milk,Vegetables"

},

{

sno:2,

title:"Prepare test",

desc: "Bio"

},

{

sno:3,

title:"Clean Cupboard",

desc: "Declutter "

}

]);

return (

<div>

Graphical user interface, text

Description automatically generated

# 2: Use Effect Hook

# Description:

It changes the side effect of the component if the component changes it or trigger some other function.

# Code Snippet:

import './App.css';

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

function App() {

const [AlpahabetType,setAlpahabetType]=useState('A')

useEffect(() => { console.log('Alphabet changed')

return()=>{ console.log("Alphabet Triggered")

}}, [AlpahabetType])

return (

<div className='Text'>

<h1>Use Effect Hook</h1>

<button className='ButtonR' onClick={()=>setAlpahabetType('A')}>A</button>

<button className='ButtonG'onClick={()=>setAlpahabetType('B')}>B</button>

<button className='ButtonB'onClick={()=>setAlpahabetType('C')}>C</button>

<h1 className='newH1'>{AlpahabetType}</h1>

</div>

)

}

export default App;

A screenshot of a computer

Description automatically generated with medium confidence

# 3: Use Memo Hook

# Description:

It can reduce the computation cost and improve the performance.

# Code Snippet:

import './App.css';

import React, { useState,useMemo} from 'react';

function App() {

const [number,setNumber] = useState(0)

const doubleNumber=useMemo(()=>{

return slowfunction(number)

},[number])

return (

<div className='Text'>

<h1>Memo Hook</h1>

<input className='ButtonG' type="number" value={number} onChange={*e* =>

setNumber(parseInt(*e*.target.value))}/>

<div>{doubleNumber}</div>

</div>

)

}

function slowfunction(*num*){

for(let i=0;i<1000000;i++){}

return *num*+2

}

export default App;

A screenshot of a computer

Description automatically generated

# Use ref Hook

# Description:

Used to render dome object

# Code Snippet:

import './App.css';

import React, {useState,useEffect,useRef} from 'react';

function App()

{

const [score,setScore]=useState('')

const prevScore=useRef('')

useEffect(()=>{ prevScore.current=score

},[score])

return (

<div className='Text'>

<h1>Use Ref Hook</h1>

<input className='ButtonG'value={score} onChange={*e*=> setScore(*e*.target.value)}/>

<div >My score is {score} and it is used to be {prevScore

.current}</div>

</div>

)

}

export default App;

A screenshot of a computer

Description automatically generated

# 5: Use call back

import './App.css';

import { useState } from 'react';

import List from './List.js'

import React, {useCallback} from 'react';

function App()

{

const [dark,setdark]=useState(false)

const theme={

backgroundColor:dark? '#333':'#FFF',

color:dark?'#FFF':'#333'

}

return (

<div style={theme} className='Text'>

<button className='ButtonR' onClick={()=>setdark(*prevDark*=>!*prevDark*)}>Change Coloro

</button>

</div>

)

}

export default App;

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated