Exp1:

Try to recreate the following pattern using HTML and CSS

```
<html>
    <head>
        <style>
            #c1{
                background-color: red;
            }
            #c2{
                background-color: yellow;
            }
            #c3{
                background-color: green;
            }
            table{
                width: 400;
                height: 400;
            }
        </style>
    </head>
    <body>
        <table>
            <tr>
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
            </tr>
            <tr>
                <td id="c2"></td>
                <td id="c3"></td>
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
                <td id="c1"></td>
            </tr>
            <tr>
                <td id="c3"></td>
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
                <td id="c1"></td>
                <td id="c2"></td>
            </tr>
            <tr>
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
            </tr>
            <tr>
                <td id="c2"></td>
                <td id="c3"></td>
```

```
                <td id="c1"></td>
                <td id="c2"></td>
                <td id="c3"></td>
                <td id="c1"></td>
            </tr>
        </table>
    </body>
</html>
```

Exp2:

Implement Drag n Drop feature in HTML5

```
<html>
    <head>
        <style>
            #div1{
                width: 400;
                height: 300;
                border: 1px solid;
            }
        </style>
        <script>
            function allowDrop(e){
                e.preventDefault();
            }
            function drag(e){
                e.dataTransfer.setData("text",e.target.id);
            }
            function drop(e){
                e.preventDefault();
                var data=e.dataTransfer.getData("text");
                e.target.appendChild(document.getElementById(data));
            }
        </script>
    </head>
    <body>
        <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
        <img id="dic" src="carrot.jpg" ondragstart="drag(event)" draggable="true">
    </body>
</html>
```

Exp3:

Demonstrate Event Bubbling with examples

```
<html>
    <body>
        <div id="outer">
            <div id="inner">
                <button id="b1">Click</button>
            </div>
        </div>
        <script>
            document.getElementById("inner").addEventListener("click",()=>{
                alert("Inner Clicked");
            })
            document.getElementById("outer").addEventListener("click",()=>{
                alert("Outer Clicked");
            })
        </script>
    </body>
</html>
```

Exp4:

Design a calculator using javascript and css

```
<html>
    <head>
        <style>
            #answer{
                width: 154;
            }
            input{
                width: 35;
            }
        </style>
    </head>
    <body>
        <table>
            <tr>
                <input type="text" id="answer">
            </tr>
            <tr>
                <td><input type="button" value="1" onclick="answer.value+='1'"></td>
                <td><input type="button" value="2" onclick="answer.value+='2'"></td>
                <td><input type="button" value="3" onclick="answer.value+='3'"></td>
                <td><input type="button" value="4" onclick="answer.value+='4'"></td>
            </tr>
            <tr>
                <td><input type="button" value="5" onclick="answer.value+='5'"></td>
                <td><input type="button" value="6" onclick="answer.value+='6'"></td>
                <td><input type="button" value="7" onclick="answer.value+='7'"></td>
                <td><input type="button" value="8" onclick="answer.value+='8'"></td>
            </tr>
            <tr>
                <td><input type="button" value="9" onclick="answer.value+='9'"></td>
                <td><input type="button" value="0" onclick="answer.value+='0'"></td>
                <td><input type="button" value="C" onclick="answer.value=''"></td>
                <td><input type="button" value="CE"
onclick="answer.value=answer.value.slice(0,-1)"></td>
            </tr>
            <tr>
```

```
                <td><input type="button" value="+" onclick="answer.value+='+'"></td>
                <td><input type="button" value="-" onclick="answer.value+='-'"></td>
                <td><input type="button" value="*" onclick="answer.value+='*'"></td>
                <td><input type="button" value="/" onclick="answer.value+='/'"></td>
            </tr>
            <tr>
                <td><input type="button" value=")" onclick="answer.value+=')'"></td>
                <td><input type="button" value="(" onclick="answer.value+='('"></td>
                <td><input type="button" value="." onclick="answer.value+='.'"></td>
                <td><input type="button" value="="
onclick="answer.value=eval(answer.value)"></td>
            </tr>
        </table>
    </body>
</html>
```

Exp5:

Demonstrate Higher order functions with examples-map(),reduce(),filter()

```
var x=[1,2,3,4,5]
var y=x.map(x=>x*2)
var z=x.filter(x=>x%2==0)
var w=x.reduce(sum,0)
function sum(value,s){
    return value+s;
}
console.log(x)
console.log(y)
console.log(z)
console.log(w)
```

Exp6:

Create a class component for counter in Reactjs

```
import React,{useState} from "react";
function App(){
    const[count,setCount]=useState(0)
    return(
        <div>
            <p>You clicked {count} times</p>
            <button onClick={()=>setCount(count+1)}>Increment</button>
            <button onClick={()=>setCount(count-1)}>Decrement</button>
        </div>
    )
}
export default App;
```

Exp7:

Create a class component for changing the color of text in Reactjs

```
import React,{useState} from "react";
function App(){
    const[iscolor,setColor]=useState('');
    return(
        <div>
            <p><span style={{color:iscolor?'green':''}}>VVIT.COM</span>Nambur</p>
            <button onClick={()=>setColor(iscolor=>!iscolor)}>Toggle Clicked</button>
        </div>
    )
}
export default App;
```

Exp8:

Create a class component for viewing an array of objects in tabular form

```
import React from "react";
function App(){
    const studentList=[
        {
            id:'501',
            name:'rajesh',
            course:'BTECH',
        },
        {
            id:'501',
            name:'rajesh',
            course:'BTECH',
        },
        {
            id:'501',
            name:'rajesh',
            course:'BTECH',
        },
        {
            id:'501',
            name:'rajesh',
            course:'BTECH',
        },
    ];
    const list=studentList.map((element)=>{
        return(
            <tr>
                <td>{element.id}</td>
                <td>{element.name}</td>
                <td>{element.course}</td>
            </tr>
        )
    });
    return(
        <div>
            <table>
                <thead>
                    <th>ID</th>
```

```
                    <th>NAME</th>
                    <th>COURSE</th>
                </thead>
                <tbody>{list}</tbody>
            </table>
        </div>
    )
}
export default App;
```

Exp9:

Display a digital clock in Reactjs

```
import React,{useState,useEffect} from "react";
function App(){
    const[date,setDate]=useState(new Date());
    useEffect(()=>{
        setInterval(()=>{
            setDate(new Date())
        },1000)
    },[])
    return(
        <div>{date.toLocaleTimeString()}</div>
    )
}
export default App;
```

Exp10:

Demonstarte useState Hook with help of sample text

```
import React,{useState} from "react";
function App(){
    const[count,setCount]=useState(0)
    return(
        <div>
            <p>You clicked {count} times</p>
            <button onClick={()=>setCount(count+1)}>Increment</button>
            <button onClick={()=>setCount(count-1)}>Decrement</button>
        </div>
    )
}
export default App;
```

Exp11:

Demonstrate useEffect Hook with example

```
import React,{useState,useEffect, useDebugValue} from "react";
const useDocumentTitle=title=>{
    useEffect(()=>{
        document.title=title;
    },[title])
}
function App(){
    const[count,setCount]=useState(0);
    useDocumentTitle(`You clicked ${count} times`);
    return(
        <div>
            <p>You clicked {count} times</p>
            <button onClick={()=>setCount(count+1)}>Increment</button>
        </div>
    )
}
export default App;
```

Exp12:

Design a BMI Calculator using Reactjs

```
import React,{useState} from "react";
function App(){
    const[height,setHeight]=useState('');
    const[weight,setWeight]=useState('');
    const[bmi,setBmi]=useState('');
    const[message,setMessage]=useState('');
    const bmicalculate=()=>{
        if(height&&weight){
        const hinm=height/100;
        const bmiv=(weight/(hinm*hinm)).toFixed(2);
        setBmi(bmiv)
        let m='';
        if(bmiv<16){
            m='severe Thinness';
        }else if(bmiv>=16 && bmiv<17){
            m='moderate Thinness';
        }else if(bmiv>=17 && bmi<18.5){
            m='Mild Thinness';
        }else if(bmiv>=18.5&&bmiv<25){
            m='Normal weight'
        }else if(bmiv>=25&&bmiv<30){
            m='Over weight'
        }else if(bmiv>=30&&bmiv<35){
            m='Obese weight I'
        }else if(bmiv>=35&&bmiv<40){
            m='Obese weight II'
        }else {
            m='Obese weight III'
        }
        setMessage(m);
    }else{
        setBmi('');
        setMessage('');
    }
}
return(
```

```
    <div>
        <h1>BMI CALCULATOR</h1>
        <div>
            <label>Enter Your Height(cm) :</label>
            <input type="number" id="height" value={height}
onChange={(e)=>setHeight(e.target.value)}/>
        </div>
        <div>
            <label>Enter Your Weight(kg) :</label>
            <input type="number" id="weight" value={weight}
onChange={(e)=>setWeight(e.target.value)}/>
        </div>
        <button onClick={bmicalculate}>Click to BMI</button>
        {bmi&&message&&(
            <div>
                <p>Your BMI:<span>{bmi}</span></p>
                <p>Result:<span>{message}</span></p>
            </div>
        )}
    </div>
)
}
export default App;
```