**EXPERIMENT-1**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Lab 1</title>

<style>

td{

width: 50px;

height: 50px;

}

.c1{

background-color: chocolate;

}

.c2{

background-color:orange;

}

.c3{

background-color: brown;

}

</style>

</head>

<body>

<Table>

<tr>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

</tr>

<tr>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

</tr>

<tr>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

</tr>

<tr>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

</tr>

<tr>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

</tr>

<tr>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

<td class="c3"></td>

<td class="c1"></td>

<td class="c2"></td>

</tr>

</Table>

</body>

</html>

**EXPERIMENT-2**

<!DOCTYPE HTML>

<html>

<head>

<script>

function allowDrop(ev) {

ev.preventDefault();

}

function drag(ev) {

ev.dataTransfer.setData("text", ev.target.id);

}

function drop(ev) {

ev.preventDefault();

var data = ev.dataTransfer.getData("text");

ev.target.appendChild(document.getElementById(data));

}

</script>

<style>

#drag1{

background-color: brown;

width: 336px;

height: 69px;

text-align: center;

align-content: center;

}

td{

width: 50%;

height: 200px;

border: 1px dotted black;

}

</style>

</head>

<body>

<Table style="width: 100%;">

<tr>

<td ondrop="drop(event)" ondragover="allowDrop(event)"></td>

<td ondrop="drop(event)" ondragover="allowDrop(event)">

<div id="drag1" draggable="true" ondragstart="drag(event)">

this box can be draged and drop</div>

</td>

</tr>

</Table>

</body>

</html>  
  
**EXPERIMENT-3**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<div id="parent">

<button>

parent

</button>

<div id = "child">

<button>

child

</button>

</div>

</div>

<a href="example2.html">example2</a>

<script>

document.getElementById('parent').addEventListener(

"click",function () {

alert("you have clicked parent");

}

)

document.getElementById('child').addEventListener(

"click", function (){

alert("you have clicked Child")

}

)

</script>

</body>

</html>  
  
  
**EXPERIMENT-4**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calculator</title>

<script>

function calc(s) {

var equationEle = document.getElementById('result')

if (s == '=') {

equationEle.value = eval(equationEle.value)

return;

}

console.log(s == 'C')

if ( s == 'C'){

equationEle.value = ""

return;

}

if ( s == 'CE'){

equationEle.value = equationEle.value.substring(0,equationEle.value.length - 1)

return;

}

equationEle.value += s

}

var a = ' "kkek" '

//"kkek"

</script>

<style>

input{

width: 50px;

height: 50px;

}

</style>

</head>

<body>

<Table>

<tr>

<td colspan="4">

<input id="result" type="text" style="width: 220px; height: 30px;">

</td>

</tr>

<tr>

<td><input type="button" value="(" onclick="document.getElementById('result').value += '('"></td>

<td><input type="button" value="CE" onclick="calc('CE')"></td>

<td><input type="button" value=")" onclick="calc(')')"></td>

<td><input type="button" value="C" onclick="calc('C')"></td>

</tr>

<tr>

<td><input type="button" value="1" onclick="calc('1')"></td>

<td><input type="button" value="2" onclick="calc('2')"></td>

<td><input type="button" value="3" onclick="calc('3')"></td>

<td><input type="button" value="+" onclick="calc('+')"></td>

</tr>

<tr>

<td><input type="button" value="4" onclick="calc('4')"></td>

<td><input type="button" value="5" onclick="calc('5')"></td>

<td><input type="button" value="6" onclick="calc('6')"></td>

<td><input type="button" value="-" onclick="calc('-')"></td>

</tr>

<tr>

<td><input type="button" value="7" onclick="calc('7')"></td>

<td><input type="button" value="8" onclick="calc('8')"></td>

<td><input type="button" value="9" onclick="calc('9')"></td>

<td><input type="button" value="\*" onclick="calc('\*')"></td>

</tr>

<tr>

<td><input type="button" value="." onclick="calc('.')"></td>

<td><input type="button" value="0" onclick="calc('0')"></td>

<td><input type="button" value="=" onclick="calc('=')"></td>

<td><input type="button" value="/" onclick="calc('/')"></td>

</tr>

</Table>

</body>

</html>  
  
**EXPERIMENT-5**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<script>

//filter

const numbers = [1,2,3,4,5,6,7,8,9]

const evenNum = numbers.filter(x => {return x%2 == 0})

document.writeln("array before filtering"+numbers+"<br>")

document.writeln("array after filtering evens numbers",evenNum,"<br>")

//reduce

const num = [1,2,3,4,5,6,7,8,9]

const sum = num.reduce((accu,curr)=> accu+curr,0)

document.writeln(`sum of ${num} using reduce is ${sum} <br>`)

//map

var nums = [1,2,3,4,5,6,7,8,9]

const doubles = nums.map((a)=> a\*2 )

document.writeln(doubles+"<br>")

//sort

nums = [2,4,6,3,7,3,7,3,6]

document.writeln("before sort",nums,"<br>")

nums.sort()

document.writeln("after sort",nums,"<br>")

//for-each

nums = [1,2,3,4,5,6,7,8]

document.writeln("using for-each <br>")

nums.forEach(element => {

document.write(element)

});

</script>

</body>

</html>  
  
**EXPERIMENT-6**

import React, { useState } from "react";

import "./App.css";

const App = () => {

const [counter, setCounter] = useState(0)

const handleClick1 = () => {

setCounter(counter + 1)

}

const handleClick2 = () => {

setCounter(counter - 1)

}

return (

<div style={{

display: 'flex',

flexDirection: 'column',

alignItems: 'center',

justifyContent: 'center',

fontSize: '300%',

position: 'absolute',

width: '100%',

height: '100%',

top: '-15%',

}}>

Counter App

<div style={{

fontSize: '120%',

position: 'relative',

top: '10vh',

}}>

{counter}

</div>

<div className="buttons">

<button style={{

fontSize: '60%',

position: 'relative',

top: '20vh',

marginRight: '5px',

}}

onClick={handleClick1}>Increment

</button>

<button style={{

fontSize: '60%',

position: 'relative',

top: '20vh',

marginLeft: '5px',

}}

onClick={handleClick2}>Decrement

</button>

</div>

</div>

)}

export default App  
  
**EXPERIMENT-7**

import {useState} from 'react';

const App = () => {

const [isColored, setIsColored] = useState(false);

return (

<div>

<p>

<span style={{color: isColored ? 'green' : ''}}>VVITGUNTUR.COM</span>

Nambur

</p>

<button onClick={() => setIsColored(isColored => !isColored)}>

Toggle isColored

</button>

</div>

);

};

export default App;

**EXPERIMENT-8**

import React from "react";

import "./App.css";

function RenderingArrayOfObjects() {

const studentlist = [

{

id: "501",

name: "Rajesh",

course:"B.TEch",

},

{

id: "502",

name: "Venkatesh",

course:"B.TEch",

},

{

id: "503",

name: "Ramesh",

course:"B.TEch",

},

{

id: "504",

name: "Suresh",

course:"B.TEch",

},

{

id: "505",

name: "Kumar",

course:"B.TEch",

},

];

const list = studentlist.map((element) => {

return (

<tr className="students" align="center">

<td>{element.id}</td>

<td>{element.name}</td>

<td>{element.course}</td>

</tr>

);

});

return (

<div>

<table align="center" border="1">

<thead>

<tr>

<th> Id</th>

<th>Name</th>

<th>Course</th>

</tr>

</thead>

<tbody>{list}</tbody>

</table>

</div>

);

}

function App() {

return (

<div className="App">

<div>

<h1 className="Students">Student List</h1>

<br></br>

<RenderingArrayOfObjects />

</div>

</div>

);

}

export default App;  
  
**EXPERIMENT-9**

import { useState, useEffect } from 'react';

function App() {

const [date, setDate] = useState(new Date());

useEffect(()=>{

setInterval(() => {

setDate(new Date())

},1000)

},[])

return (

<p className='DigitalClock'>

{date.toLocaleTimeString()}

</p>

);

}

export default App;

**EXPERIMENT-10**

import React, { useState } from 'react';

function Hook() {

const [count, setCount] = useState(0);

return (

<div>

<p>You clicked {count} times</p>

<button onClick={() => setCount(count + 1)}>

Click me

</button>

</div>

);

}

export default Hook;  
  
**EXPERIMENT-12**

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

const useDocumentTitle = title => {

useEffect(() => {

document.title = title;

}, [title])

}

function CustomCounter() {

const [count, setCount] = useState(0);

const incrementCount = () => setCount(count + 1);

useDocumentTitle(`You clicked ${count} times`);

return (

<div>

<p>You clicked {count} times</p>

<button onClick={incrementCount}>Click me</button>

</div>

)

}

export default CustomCounter;  
  
**EXPERIMENT-14**

import React,{useState} from 'react';

import './App.css';

function BmiCalculator() {

const [heightValue, setHeightValue] = useState('');

const [weightValue, setWeightValue] = useState('');

const [bmiValue, setBmiValue] = useState('');

const [bmiMessage, setBmiMessage] = useState('');

const calculateBmi = () => {

if (heightValue && weightValue) {

const heightInMeters = heightValue / 100;

const bmi = (weightValue / (heightInMeters \* heightInMeters)).toFixed(2);

setBmiValue(bmi);

let message = '';

if (bmi < 16) {

message = 'Severe Thinness ';

}else if (bmi >= 16 && bmi <=17) {

message = 'Moderate Thinness';

}

else if (bmi >= 17 && bmi <=18.5) {

message = 'Mild Thinness ';

}

else if (bmi >= 18.5 && bmi < 25) {

message = 'Normal weight';

} else if (bmi >= 25 && bmi < 30) {

message = 'Overweight';

}else if (bmi >= 30 && bmi < 35) {

message = 'Obese Class I ';

}

else if (bmi >= 35 && bmi < 40) {

message = 'Obese Class II ';

}

else {

message = 'Obese Class III';

}

setBmiMessage(message);

} else {

setBmiValue('');

setBmiMessage('');

}

};

return (

<div className="container">

<h1>BMI Calculator</h1>

<div className="input-container"><label htmlFor="height">Enter Your Height (cm):</label>

<input

type="number"

id="height"

value={heightValue}

onChange={(e) => setHeightValue(e.target.value)}

/>

</div>

<div className="input-container">

<label htmlFor="weight">Enter Your Weight (kg):</label>

<input

type="number"

id="weight"

value={weightValue}

onChange={(e) => setWeightValue(e.target.value)}

/>

</div>

<button className="calculate-btn" onClick={calculateBmi}>

Click to Calculate BMI

</button>

{bmiValue && bmiMessage && (

<div className="result">

<p> Your BMI:

<span className="bmi-value">{bmiValue}</span>

</p>

<p>Result:

<span className="bmi-message">{bmiMessage}</span>

</p>

</div>

)}

</div>

);

}

export default BmiCalculator;