1. Try to recreate the following patterns using HTML and CSS only.

```
<html>
 <head>
     <title>colors</title>
 </head>
 <body>
     <style>
         table{
             text-align: center;

 }
     </style>
     <table align="center" cellpadding="11" cellspacing="2">
         <tr>
             <td bgcolor="red"></td><pre> </pre>
             <td bgcolor="blue"></td><pre> </pre>
             <td bgcolor="orange"></td><pre> </pre>
             <td bgcolor="red"></td><pre> </pre>
             <td bgcolor="red"></td><pre> </pre>
             <td bgcolor="red"></td><pre> </pre>
         </tr>
         <tr>
             <td bgcolor="blue"></td><pre> </pre>
             <td bgcolor="blue"></td><pre> </pre>
             <td bgcolor="blue"></td><pre> </pre>
             <td bgcolor="blue"></td><pre> </pre>
             <td bgcolor="blue"></td><pre> </pre>
             <td bgcolor="blue"></td><pre> </pre>
         </tr>
         <tr>
             <td bgcolor="green"></td><pre> </pre>
             <td bgcolor="green"></td><pre> </pre>
             <td bgcolor="green"></td><pre> </pre>
             <td bgcolor="green"></td><pre> </pre>
             <td bgcolor="green"></td><pre> </pre>
             <td bgcolor="green"></td><pre> </pre>
         </tr>
         <tr>
             <td bgcolor="yellow"></td><pre> </pre>
             <td bgcolor="yellow"></td><pre> </pre>
             <td bgcolor="yellow"></td><pre> </pre>
             <td bgcolor="yellow"></td><pre> </pre>
             <td bgcolor="yellow"></td><pre> </pre>
             <td bgcolor="yellow"></td><pre> </pre>
         </tr>
         <tr>
             <td bgcolor="violet"></td><pre> </pre>
             <td bgcolor="violet"></td><pre> </pre>
             <td bgcolor="violet"></td><pre> </pre>
             <td bgcolor="violet"></td><pre> </pre>
             <td bgcolor="violet"></td><pre> </pre>
             <td bgcolor="violet"></td><pre> </pre>
         </tr>
         <tr>
             <td bgcolor="black"></td><pre> </pre>
             <td bgcolor="black"></td><pre> </pre>
             <td bgcolor="black"></td><pre> </pre>
             <td bgcolor="black"></td><pre> </pre>
             <td bgcolor="black"></td><pre> </pre>
```

```
                <td bgcolor="black"></td><pre> </pre>
            </tr>
            </table></center>
    </body>
</html>
```

----

2. Implement Drag n Drop feature in HTML 5

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <title>DragNDrop Demo</title>
 <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
 <script src="../js/main.js"></script>
 <div id="container-1" ondragover="allowDrop(event)"
ondrop="drop(event)">
 <div id="image" draggable="true" ondragstart="drag(event)">
 </div>
 </div>
 <div id="container-2" ondragover="allowDrop(event)"
ondrop="drop(event)">
 </div>
</body>
</html>
```

Styles.css:

```
body{
 width: 50%;
 margin: 0 auto;
 background-color: salmon;
}
#container-1, #container-2{
 display: inline-block;
 width: 320px;
 height: 320px;
 border: 0.2em solid black;
 margin: 50px 10px;
 background-color: whitesmoke;
 border-radius: 10px;
}
#image:hover{
 border: 0.4em dashed blue !important;
}
#image{
 background: url('https://source.unsplash.com/random/300x300');
 position: relative;
 height: 300px;
 width:300px;
 top:5px;
 left:5px;
 border:5px solid #ccc;
 border-radius: 5px;
 cursor: pointer;
}
```

```
------------------------------------------------------------------
Main.js:

function drag(e){
 e.dataTransfer.setData('image',e.target.id);
}
function allowDrop(e){
 e.preventDefault();
}
function drop(e){
 e.preventDefault();
 var data = e.dataTransfer.getData('image');
 e.target.appendChild(document.getElementById(data));
-----------------------------------------------------------------
```

3. Demonstrate Event bubbling with necessary examples

```
   <!DOCTYPE html>
<html>
<head>
      <meta charset="utf-8">
      <meta name ="viewport" content="width=device-width">
      <title>Event Bubbling</title>

</head>

<body>
      <div id="parent">
            <button id="child" onclick="event.stopPropagation()">Child</button>
      </div>
      <script>
            var parent = document.querySelector('#parent');
            var child = document.querySelector('#child');

            parent.addEventListener('click', function() {
                  console.log("Parent Clicked");
            });

            child.addEventListener('click', function(){
                  console.log("Child Clicked");
            });
      </script>
</body>
</html>
-----------------------------------------------------------------------------------
---
```

4.Design a Calculator using Java script and relevant CSS.

```
  <html>
    <head>
        <title>Calcultor</title>

        <style>
            table{
                padding: 10px;
                border-radius: 1em;
                height: 380px;
                width: 400px;
                margin: auto;
                box-shadow: rgba(0, 0, 0, 0.19) 0px 10px 20px, rgba(0, 0, 0, 0.23)
0px 6px 6px;
            }
            input[type=button] {
                font-family: 'Orbitron', sans-serif;
                color: white;
                border: solid black 0.5px;
                width: 100%;
                border-radius: 5px;
                height: 70%;
                color:black;
            }
            input[type=text]{
                border: solid black 0.1px;
                width:100%;
                height:100%;
            }
        </style>
    </head>
    <body>
        <form align="center">
            <table align="center">
                <tr>
                    <td colspan="4"><input type='text' style="background-
color:gray" id="inp" disabled/></td></td>
                </tr>
                <tr>
                    <td>
                        <input type="button" value=" ( " onclick="display('(')"/>
                    </td>
                    <td>
                        <input type="button" value=" CE " onclick="delee()"/>

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

                    </td>
                </tr>
                <tr>
                    <td>
                    <input type="button" value=" 1 " onclick="display('1')"/>
                    </td>
                    <td>
                        <input type="button" value=" 2 " onclick="display('2')"/>

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

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

                    </td>
                </tr>
                <tr>
                    <td>
                        <input type="button" value="4" onclick="display('4')"/>
                    </td>
                    <td>
                        <input type="button" value="5" onclick="display('5')"/>

                    </td>
                    <td>
                        <input type="button" value="6" onclick="display('6')"/>
                    </td>
                    <td>
                        <input type="button" value="-" onclick="display('-')"/>
                    </td>
                </tr>
                <tr>
                    <td>
                        <input type="button" value="7" onclick="display('7')"/>
                    </td>
                    <td>
                        <input type="button" value="8" onclick="display('8')"/>
                    </td>
                    <td>
                        <input type="button" value="9" onclick="display('9')"/>
                    </td>
                    <td>
                        <input type="button" value="*" onclick="display('*')"/>

                    </td>
                    </tr>
                    <tr>
                        <td>
                            <input type="button" value="." onclick="display('.')"/>
                        </td>
                        <td>
                            <input type="button" value="0" onclick="display('0')"/>
                        </td>
                        <td>
                            <input type="button" value="=" onclick="calculate()"/>
                        </td>
                        <td>
                            <input type="button" value="/" onclick="display('/')"/>
                        </td>
                    </tr>


                </tr>
            </table>
        </form>
        <script>
            function clearScreen(){
                document.getElementById("inp").value ="";
            }
            function display(value){
                document.getElementById("inp").value += value;
            }
            function calculate(){
                var p = document.getElementById("inp").value;
                var q = eval(p);
                document.getElementById("inp").value = q;
            }
            function delee(){
                var st=document.getElementById("inp").value;
                document.getElementById("inp").value=st.slice(0,st.length-1);
            }
        </script>
    </body>
</html>
```

5.Demonstrate Higher order functions with necessary examples – filter(), reduce() and map()

  Map, Filter, Reduce

App.js:

```
import './App.css';
import MyCart from './components/MyCart';
function App() {
 return (
 <div>
 <MyCart />
 </div>
 );
}
export default App;
```

----

data.js:

```
export const products = [
 {
 id: 59,
 title: "Spring and summershoes",
 price: 20,
 quantity: 3,
 total: 60,
 discountPercentage: 8.71,
 discountedPrice: 55,
 },
 {
 id: 88,
 title: "TC Reusable Silicone Magic Washing Gloves",
 price: 29,
 quantity: 2,
 total: 58,
 discountPercentage: 3.19,
 discountedPrice: 56,
 },
 {
 id: 18,
 title: "Oil Free Moisturizer 100ml",
 price: 40,
 quantity: 2,
 total: 80,
 discountPercentage: 13.1,
 discountedPrice: 70,
 },
 {
 id: 95,
 title: "Wholesale cargo lashing Belt",
 price: 930,
 quantity: 1,
 total: 930,
 discountPercentage: 17.67,
 discountedPrice: 766,
 },
 {
 id: 39,
 title: "Women Sweaters Wool",
 price: 600,
 quantity: 2,
 total: 1200,
 discountPercentage: 17.2,
 discountedPrice: 994,
 },
];
------------------------------------------------------------
Helper.js
export const findSum = (array) => {
 const totalSum = array.reduce((acc, item) => {
 return acc + item.discountedPrice;
 }, 0);
 return totalSum;
};
-----------------------------------------------------------
MyCart.js:

import React, { Component } from "react";
import DisplayCart from "./DisplayCart";
import FilterData from "./FilterData";
import "./styles.css"
export class MyCart extends Component {
 constructor(props) {
 super(props);
 this.state = {
 displayCart: false,
 displayFilter: false,
 };
 }
 handleClick = e => {
 if(e.target.id === 'show'){
 this.setState({displayCart: true , displayFilter: false})
 }
 else if(e.target.id === 'filter'){
 this.setState({displayFilter: true, displayCart: false})
 }
 }
 render() {
 return <div>
 <h1>My Shopping Cart</h1>
 <div className="box">
 <button className='myButton'
 id='show' onClick={this.handleClick}>Show Cart</button>
 <button className='myButton'
 id='filter' onClick={this.handleClick}>Filter Cart</button>
 <div className='container'>
 {this.state.displayCart && <DisplayCart />}
 {this.state.displayFilter && <FilterData />}
 </div>
 </div>
 </div>;
 }
}
export default MyCart;
--------------------------------------------------------
DisplayCart.js:

import React, { Component } from 'react'
import {products as data} from '../data/data.js'
import { findSum } from './helper.js';
export class DisplayCart extends Component {
 constructor(props) {
 super(props)

 this.state = {
 totalSum: 0,
 isClicked: false,
 }
 }
 handleClick = () => {
 const total = findSum(data);
 this.setState({totalSum: total, isClicked: !this.state.isClicked});
 }
 render() {
 return (
 <div>
 <table border={1}>
 <tr>
 <th>Name of Item</th>
 <th>Price of Item</th>
 <th>Quantity </th>
 <th>Total </th>
 <th>Discount Percentage </th>
 <th>Discount Price </th>
 </tr>
 {
 data.map((item)=>{
 return (
 <tr key={item.id}>
 <td>{item.title}</td>
 <td>{item.price}</td>
 <td>{item.quantity}</td>
 <td>{item.total}</td>
 <td>{item.discountPercentage}</td>
 <td>{item.discountedPrice}</td>
 </tr>
 )
 })
 }
 </table>
 <button onClick={this.handleClick}>FindSum</button>
 {this.state.isClicked && this.state.totalSum}
 </div>
 )
 }
}
export default DisplayCart;
-----------------------------------------------------
FilterData.js:
```

```
import { Component } from "react";
import {products} from '../data/data.js';
class FilterData extends Component{
 render(){
 const filteredData = products.filter((item)=>{
 return item.quantity <=2 ? item : '';
 });
 console.log(filteredData);
 return (
 <div>
 <table border={1}>
 <tr>
 <th>Name of Item</th>
 <th>Price of Item</th>
 <th>Quantity </th>
 </tr>
 {
 filteredData.map((item)=>{
 return (
 <tr>
 <td>{item.title}</td>
 <td>{item.price}</td>
 <td>{item.quantity}</td>
 </tr>
 )
 })
 }
 </table>
 </div>
 );
 };
}
export default FilterData;
-------------------------------------------------------------------------
```

6.Create a Class Component for Counter in React JS
  Counter App (Cric Score)

App.js:

```
import './App.css';
import CricScore from './components/CricScore';
function App() {
 return (
 <div>
 <CricScore />
 </div>
 );
}
export default App;
---------------------------------------------------------
```

CricScore.js:

```
import React, { Component } from "react";
export class CricScore extends Component {
 constructor(props) {
 super(props);
 this.state = {
 score: 0,
 wickets: 0,
 balls: 0,
 overs: 0,
 };
 }
 changeBalls = () => {
 if ((this.state.balls + 1) % 6 === 0) {
 this.setState((prevState) => ({
 ...prevState,
 overs: this.state.overs + 1,
 }));
 }
 this.setState((prevState) => ({
 ...prevState,
 balls: (this.state.balls + 1) % 6,
 }));
 };
 changeScore = (value, countBall) => {
 this.setState((prevState) => ({
 ...prevState,
 score: this.state.score + value,
 }));
 countBall && this.changeBalls();
 };
 handleClick = (e) => {
 const id = e.target.id;
 switch (id) {
 case "2":
 this.changeScore(2);
 break;
 case "3":
 this.changeScore(3);
 break;
 case "4":
 this.changeScore(4);
 break;
 case "6":
 this.changeScore(6);
 break;
 case "1":
 this.changeScore(1);
 break;
 case "No Ball":
 this.changeScore(1,false);
 break;
 case "Wide":
 this.changeScore(1,false);
 break;
 case "Wicket":
 this.setState(prevState => ({...prevState, wickets:
this.state.wickets + 1}));
 break;
 case "0":
 this.changeBalls();
 break;
 default:
 break;
 }
 };
 render() {
 return (
 <div style={{ textAlign: "center" }}>
 <h1>Score Board : Cricket</h1>
 <h2>
 Score : {this.state.score} / {this.state.wickets}
 </h2>
 <h2>
 Overs : {this.state.overs}.{this.state.balls}{" "}
 </h2>
 <div>
 <button id="0" onClick={this.handleClick}>
 0
 </button>
 <button id="1" onClick={this.handleClick}>
 1
 </button>
 <button id="2" onClick={this.handleClick}>
 2
 </button>
 <button id="3" onClick={this.handleClick}>
 3
 </button>
 <button id="4" onClick={this.handleClick}>
 4
 </button>
 <button id="6" onClick={this.handleClick}>
 6
 </button>
 <button id="Wide" onClick={this.handleClick}>
 Wide
 </button>
 <button id="No Ball" onClick={this.handleClick}>
 No Ball
 </button>
 <button id="Wicket" onClick={this.handleClick}
disabled={this.state.wickets>9? true : false}>
 Wicket
 </button>
 </div>
 </div>
 );
 }
}
export default CricScore;
-------------------------------------------------------------------
```

7.Create a Class component for Changing the color of the text given in React JS

ChangeColors.js:

```
import React, { Component } from "react";
import "./styles.css";
export class ChangeColors extends Component {
 constructor(props) {
 super(props);
 this.state = {
 showRed: true,
 showYellow: false,
 showGreen: false,
 };
 }
 componentDidMount() {
 setTimeout(() => {
 this.setState({ showRed: false, showYellow: true, showGreen:
false });
 }, 4000);
 }
 componentDidUpdate(prevProps, prevState) {
 console.log("PrevProps:", prevProps);
 console.log("PrevState:", prevState);
 console.log("CurrentProps:", this.props);
 console.log("CurrentState:", this.state);
 if (this.state.showYellow) {
 setTimeout(() => {
 this.setState({ showRed: false, showYellow: false, showGreen:
true });
 }, 2000);
 }
 if(this.state.showGreen){
 setTimeout(()=>{
 this.setState({showRed: true, showYellow: false, showGreen:
false})
 },4000);
 }
 if(this.state.showRed){
 setTimeout(() => {
 this.setState({ showRed: false, showYellow: true,
showGreen: false });
 }, 4000);
 }
 }
 render() {
 return (
 <div>
 <div className={this.state.showRed ? "circle-red" : "circle"}
/>
 <div className={this.state.showYellow ? "circle-yellow" :
"circle"} />
 <div className={this.state.showGreen ? "circle-green" :
"circle"} />

 </div>
 );
 }
}
export default ChangeColors;
----------------------------------------------------------------------
Styles.css:

.circle {
 border: 1px solid black;
 border-radius: 50%;
 width: 100px;
 height: 100px;
 background-color: white;
 transition: background-color 0.5s ease-in-out;

}
```

```css
.circle-red {
 border: 1px solid red;
 border-radius: 50%;
 width: 100px;
 height: 100px;
 background-color: red;
 transition: background-color 0.5s ease-in-out;
 outline: none;
}
.circle-green {
 border: 1px solid green;
 border-radius: 50%;
 width: 100px;
 height: 100px;
 background-color: green;
 transition: background-color 0.5s ease-in-out;
 outline: none;
}
.circle-yellow {
 border: 1px solid orange;
 border-radius: 50%;
 width: 100px;
 height: 100px;
 background-color: orange;
 transition: background-color 0.5s ease-in-out;
 outline: none;
}
.line{
 border: 1px solid black;
 height: 100px;
 width: 20px;
}
---------------------------------------------------------------
```

App.js:

```javascript
import './App.css';
import ChangeColors from './components/ChangeColors.js/ChangeColors';
function App() {
 return (
 <div style={{width:'90%', margin: '0 auto'}}>
 <ChangeColors />
 </div>
 );
}
export default App;
----------------------------------------------------------------------------------
```

8.Class a Class Component for viewing an array of objects in a tabular form.

DisplayData.js:

```javascript
import React, { Component } from "react";
import { products } from "../data";
export class DisplayData extends Component {
 render() {
 return (
 <div>
 <table border={1}>
 <tr>
 <th>id</th>
 <th>Title</th>
 <th>Price</th>
 <th>Quantity</th>
 </tr>
 {products.map((item) => {
 return (
 <tr key={item.id}>
 <td>{item.id}</td>
 <td>{item.title}</td>
 <td>{item.price}</td>
 <td>{item.quantity}</td>
 </tr>
 );
 })}
 </table>
 </div>
 );
 }
}
export default DisplayData;
-----------------------------------------------------
data.js:

export const products = [
 {
 id: 1,
 title: "Spring and summershoes",
 price: 20,
 quantity: 3,
 total: 60,
 discountPercentage: 8.71,
 discountedPrice: 55,
 },
 {
 id: 2,
 title: "TC Reusable Silicone Magic Washing Gloves",
 price: 29,
 quantity: 2,
 total: 58,
 discountPercentage: 3.19,
 discountedPrice: 56,
 },
 {
 id: 3,
 title: "Oil Free Moisturizer 100ml",
 price: 40,
 quantity: 2,
 total: 80,
 discountPercentage: 13.1,
 discountedPrice: 70,
 },
 {
 id: 4,
 title: "Wholesale cargo lashing Belt",
 price: 930,
 quantity: 1,
 total: 930,
 discountPercentage: 17.67,
 discountedPrice: 766,
 },
 {
 id: 5,
 title: "Women Sweaters Wool",
 price: 600,
 quantity: 2,
 total: 1200,
 discountPercentage: 17.2,
 discountedPrice: 994,
 },
 ];
----------------------------------------------------------
 App.js:

import './App.css';
import DisplayData from './components/DisplayData';
function App() {
 return (
 <div>
 <DisplayData />
 </div>
 );
}
export default App
------------------------------------------------------------------------
```

9.Display a digital clock in React JS.

DigitaClock.js:

```
import React, { Component } from "react";
export class DigitalClock extends Component {
 constructor(props) {
 super(props);
 this.state = {
 time: "",
 };
 }
 componentDidMount(){
 this.tick();
 }
 tick = () => {
 const hours = new Date().getHours();
 const minutes = new Date().getMinutes();
 const seconds = new Date().getSeconds();
 const updatedTime = `${hours}:${minutes}:${seconds}`;
 this.setState({time: updatedTime})
 };
 componentDidUpdate(prevProps,prevState){
 if(this.state.time !== prevState.time){
 this.interval = setInterval(()=>{
 this.tick();
 },1000);
 }
 }
 componentWillUnmount(){
 clearInterval(this.interval);
 }
 render() {
 return <div>{this.state.time}</div>;
 }
}
export default DigitalClock;
------------------------------------------------------
```

App.js:

```
import './App.css';
import DigitalClock from './components/DigitalClock/DigitalClock';
function App() {
 return (
 <div style={{width:'90%', margin: '0 auto'}}>
 <DigitalClock />
 </div>
 );
}
export default App;
-------------------------------------------------------------
```

10. Demonstrate useState Hook with the help sample text.

```
//Preparation of a TODO list.

import React, {useState} from 'react'

function ViewItems() {
  const [item, setItem] = useState("");
  const [items, setItems] = useState([]);
  const handleAddItem = () => {
    setItems([...items, {"item":item, isStriked:false}]);
    setItem("");
  };
  const handleDelete = (item) => {
    const fitems = items.filter((it) => it.item !== item.item);
    setItems([...fitems]);
  };
  const handleText=(item)=>{
    const fitems = items.map((it) => {
      if(it.item === item.item){
        return {
          ...it,
          isStriked:!it.isStriked
        }
    } else {
      return it;
    }});
    setItems([...fitems]);  }

  console.log(items);
  const tdata = items.map((it, index) => (
    <tr key={index}>
      <td><span onClick={() =>handleText(it)}
      style={{textDecoration:it.isStriked?
"line-through":"none"}}>{it.item}</span></td>
      <td>
        <button onClick={() => handleDelete(it)}>X</button>
      </td>
    </tr>
  ),[]);
  return (
    <div>
      <input
```

```
        type="text"
        size="20"
        value={item}
        onChange={(e) => setItem(e.target.value)}
      />
      <button onClick={() => handleAddItem()}>Add</button>
      <h1>Items</h1>
      <table border=" ">{tdata}</table>
    </div>
  );
}

export default ViewItems;
-------------------------------------------------------
```

11. Demonstrate useContext Hook with necessary example.

```
//useContext – Demo – array of objects
```

App.js

```
import "./styles.css";
import React from 'react';
import BlogContextDemo from ‘. /BlogContextDemo'

const blogInfo = {
  React: {
    post: "Learn useContext Hooks",
    author: "Varun K"
  },
  NodeJS: {
    post: "Node Commands",
    author: "Veena M"
  }
};

export const BlogContext  = React.createContext(blogInfo);

export default function App() {
  return (
    <div className="App">
      <div>
      <h1>Hello KP</h1>
      <BlogContext.Provider value={blogInfo}>
           <BlogContextDemo />
      </BlogContext.Provider>
      </div>
    </div>
  );
}
```

BlogContextDemo.js

```
import React, {useContext} from "react";
import {BlogContext} from './App';

function BlogContextDemo() {
  const binfo = useContext(BlogContext);
  return (
      <div>
       <p>Topic: {binfo.React.post}</p>
       <p>Author: {binfo.React.author}</p>
      </div>
  );
}

export default BlogContextDemo;
```

----

12. Demonstrate useEffect Hook with necessary example.

```
//Digital Clock using useEffect()

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

function DigiClock() {

    const [mytime, getMytime] = useState(" ");

    const tick = () => {
        let time = new Date().getHours()+":"+new Date().getMinutes()+":"+new
Date().getSeconds();
        getMytime(time);
    }

    useEffect(() => {
        const t = setInterval(tick,1000);

        return () => {
            clearTimeout(t);
        }
    },[mytime])

  return (
    <div>
        <h1>DigitalClock</h1>
        <h2>{mytime}</h2>
    </div>
  )
}

export default DigiClock;
```