Internet Programming Unit 4

ReactJS Programs

React Programs, Props and Components

1. Basic React App for HelloWorld

```
import React from "react";
export default function App(){
  return(
  <div>
    <h1>Hello World!</h1>
  </div>
  )
}
```

2. Write a program to initialize an array of food items, and to randomly choose one and display it. (Objective: Variables in HTML Tags)

```
import React from "react";

export default function App(){
  return(
  <div>
    <h1>Hello World!</h1>
  </div>
  )
}
```

3. **Create a To Do List with the following features:**
    i. **A list of items**
    ii. **The heading of the list should be <person name>'s To Do List on <Day of the week>**
    iii. **Use an object to store both the person's name and the day of the week value.**

```
import React from "react"
const date= new Date();

const personDetails={
  name: "Pooja Premnath",
  theme: {
    color: "pink",
    backgroundColor: "black"
  }
}
function findDay(date){
  const day=date.getDay();
  switch(day){
    case 0: return "Sunday";
    case 1: return "Monday";
    case 2: return "Tuesday";
    case 3: return "Wednesday";
    case 4: return "Thursday";
    case 5: return "Friday";
    case 6: return "Saturday"
  }
}

export default function App(){
```

```
return(
<div style={personDetails.theme}>
  <h1>{personDetails.name}'s To Do List on
{findDay(date)}</h1>
  <ul>
    <li>Eat food</li>
    <li>Study Internet Programming </li>
    <li>Work out ToC Practice Sums</li>
  </ul>

</div>
)
```

4. Show the usage of props by creating a Student.js program that displays a student's name and email as retrieved from App.js

App.js

```
import React from "react";
import Student from './Student';

export default function App(){
  return(
    <div>
      <Student name={"Pooja"}
email={"pooja2110152@ssn.edu.in"}/>
      <Student name={"Pranav"}
email={"pranav2110152@ssn.edu.in"}/>
    </div>
  )
}
```

Student.js

```
import React from "react";


export default function Student(props){
    return(
        <div>
        <h1>Hello {props.name}</h1>
        <h1>Your email is {props.email}</h1>
        </div>
    )
}
```

5. **Write a program that stores an array of student objects with their id and name in App.js. Use props and the Map functionality to return a list of all the items in the list in Student.js**

App.js

```
import React from 'react';
import Student from './Student';


const details=[
  {id: "2110152",
  name: "pooja"
},


{id:"2110153",
name: "pranav"},


{id: "2110154",
```

```
name: "premnath"}



]
export default function App(){
  return(
    <Student studentDetails={details}/>
  )
}
```

Student.js

```
import React from 'react';
/*syntax is
props.idvariablefromapp.map(mapVar=>{return(<li>{mapVar.object
feature}</li>)})*/

export default function Student(props){
    return(
        <div>
            <ul>
            {props.studentDetails.map(mapVar=> {return
(<li>{mapVar.name} - {mapVar.id}</li>)})}
            </ul>
        </div>
    )
}
```

6. **Create a react program with a parent component (App.js) and two child components- Child1.js and Child2.js. Render the child components in the parent component.**

App.js

```
import React from "react";
import Child1 from "./Child1";
import Child2 from "./Child2";
export default function App(){
  return(
    <div>
      <Child1/>
      <Child2/>
    </div>
  )
}
```

Child1.js

```
import React from "react";
export default function Child1(){
    return(
        <div>
            <h1>Child 1</h1>
            <p>Child 1 paragraph</p>
        </div>
    )
}
```

Child2.js

```
import React from "react";
export default function Child1(){
    return(
        <div>
```

```
            <h1>Child 2</h1>
            <p>Child 2 paragraph</p>
        </div>
    )
}
```

7. Create a react application that displays student details from an array of objects, in a flex display. The student details should be retrieved and styled in Student.js

App.js

```
import React from "react";
import Student from "./Student";
const studentArray=[
  {id:"1",name: "pooja", email: "pooja2110152@ssn.edu.in"},
  {id:"2",name: "preetha", email:
"preetha2110152@ssn.edu.in"},
  {id:"3",name: "pranav", email: "pranav2110152@ssn.edu.in"},
  {id:"4",name: "premnath", email:
"premnath2110152@ssn.edu.in"},
  {id:"5",name: "rita", email: "rita2110152@ssn.edu.in"},
  {id:"6",name: "sita", email: "sita2110152@ssn.edu.in"},
  {id:"7",name: "geetha", email: "geetha2110152@ssn.edu.in"}
]
export default function App(){
  return(
    <Student studentDetails={studentArray}/>
  )
}
```

Student.js

```
import React from "react";
```

```
export default function Student(props){
    return(
        <div id="main">

        {props.studentDetails.map(mapVar=>{return (<div
id="element"> {mapVar.id} <br/> {mapVar.name} <br/>
{mapVar.email}</div>)})}
        </div>
    )
}
```

Index. css

```
#main{
  display: flex;
  flex-wrap: wrap;
}

#element{
  border: 20px orange solid;
  font-size: large;
  padding: 50px;
  margin: 50px;
}
```

8. **Write a program that uses props to display images and a description about them.**
   App.js

```
import React from "react";
import Image from "./Image";
import cat1 from './cat1.jpeg';
```

```
import cat2 from './cat2.jpg';


export default function App(){
  return(
    <div>
      <Image title="Image 1" image={cat1}/>
      <Image title="Image 2" image={cat2}/>
    </div>
  )
}
```

Image.js

```
import React from 'react';
export default function Image(props){
    return(
        <div>
            <h1>This is {props.title}</h1>
            <img src={props.image} alt="tempstring"
height="150" width="150"/>
        </div>
    )
}
```

UseState Programs

What Is 'State' in ReactJS? The state is a built-in React object that is used to contain data or information about the component. A component's state can change over time; whenever it changes, the component re-renders.
What is the useState Hook? useState is **React Hook that allows you to add state to a functional component.**

9. **Write a program to implement an up counter and a down counter using useState.**

App.js

```
import React from 'react';
import {useState} from 'react';

export default function App(){
  const [counter, setCount]=useState(0);
  function increment(){
    setCount(counter+1);
  }
  function decrement(){
    setCount(counter-1);
  }
    return(
  <div>
    <h1>Counter Value: {counter}</h1>
    <button onClick={increment}>Increment</button>
    <button onClick={decrement}>Decrement</button>
  </div>)
}
```

10. Demonstrate stateless and stateful components in React.

StateExample.js – Stateless Component

```
import React from "react";
import {useState } from "react";
 function StateExample(){
const user=”React”
   function changeName(){
```

user=”JS”; → user will not be changed to “JS” even when the button is clicked. But in console, we can see JS

```
    console.log(user);
  }
   return(
       <>
       <h1> {user}</h1>
       <button onClick={changeName}>Change Name</button>
       </>
   );
}
export default StateExample;
```

StateExample.js – Stateful component

```
import React from "react";
import {useState } from "react";
 function StateExample(){
   const [user,setUser]=useState("React");
  function changeName(){
    setUser("JS");
    console.log(user);
  }
   return(
       <>
       <h1> {user}</h1>
       <button onClick={changeName}>Change Name</button>
       </>
   );
}
```

11. Write a program that shows the state update of a all the values in an object. Consider an object with the name and id. Obtain user input to change the values using useState.

App.js

```
import React from "react";
import {useState} from "react";

const student={id: 1, name: "pooja"};

export default function App(){
  const [student1,updateStudentState]=useState(student);
function updateDetails(){
  const newId=document.getElementById("studId").value;
  const newName=document.getElementById("studName").value;
  updateStudentState({id: newId, name: newName})

}



return(
  <div>
    <h1>Student ID: {student1.id}</h1>
    <h1>Student Name: {student1.name}</h1>
    <br/>
    <input type="number" id="studId" placeholder="Input new
ID"/>
    <input type="text" id="studName" placeholder="Input new
name"/>
```

```
    <button onClick={updateDetails}>Update Student
Details</button>


  </div>
)
}
```

12. Write a program that shows the update of values in an array of objects. Validate the id of a student to change the value of the student name.

```
import React, { useState } from 'react';
export default function App() {
  const initialStudentDetails = [
    { id: 1, name: 'pooja' },
    { id: 2, name: 'rita' },
    { id: 3, name: 'pranav' }
  ];

  const [studentDetails, setStudentDetails] =
useState(initialStudentDetails);
  const [student, updateStudent] = useState({ id: '', name: ''
});

  const handleChanges = (e) => {
    const { name, value } = e.target;
    updateStudent((prevState) => ({
      ...prevState,
      [name]: value
    }));
  };

  const updateStudentDetails = () => {
```

```
    setStudentDetails((prevDetails) =>
      prevDetails.map((studentDetail) =>
        studentDetail.id === parseInt(student.id) ? {
...studentDetail, name: student.name } : studentDetail
      )
    );
  };

  return (
    <div>
      <h1>Student Details</h1>
      <input
        type="text"
        name="id"
        placeholder="Enter student id"
        onChange={handleChanges}
        value={student.id}
      />
      <input
        type="text"
        name="name"
        placeholder="Enter student name"
        onChange={handleChanges}
        value={student.name}
      />
      <button type="submit" onClick={updateStudentDetails}>
        Update Details
      </button>

      <br />
```

```
      <br />
      <br />

      <h1>Display Details</h1>
      {studentDetails.map((mapVar) => (
        <ul key={mapVar.id}>
          <li>Id: {mapVar.id}</li>
          <li>Name: {mapVar.name}</li>
        </ul>
      ))}
    </div>
  );
}
```

useReducer Programs (Refactoring of ReactJS)

13. Write a program that performs increments and decrements a counter using useReducer.

```
import React from 'react';
import { useReducer } from 'react';

var initialCount=0;

//the name of the function is the first var on the RHS
function reducer(count, action){

  switch(action.type){
    case 'increment': return count+1;
```

```
    case 'decrement': return count-1;
    case 'reset': return 0;
  }
}

//print the original value by using the og var on the LHS
export default function App(){
  const [counter, dispatch]=useReducer(reducer, initialCount);
  return(
    <div>
        <h1>Counter: {counter}</h1>
        <button onClick={()=>
dispatch({type:'increment'})}>Increment</button>
        <button onClick={()=> dispatch({type:
'decrement'})}>Decrement</button>
        <button onClick={()=>
dispatch({type:'reset'})}>Reset</button>
    </div>
  )
}
```

14. Write a program using useReducer to update the parameters of an object. For a student object, update the student name, and increment or decrement the student age.

```
//modifying a object using useReducer

import React from 'react';
import { useReducer} from 'react';

var studentObject={name:"pooja",age:20};
```

```
function changeObject(state, action){
  switch(action.type){
    case "changeName": return {...state, name: action.name}
    case "increment age": return {...state, age: state.age+1}
    case "decrement age": return {...state, age: state.age-1}
  }
}
export default function App(){
  let [ogStudent, dispatch]=useReducer(changeObject,
studentObject)

  return(
    <div>
        <h1> Student Name: {ogStudent.name} </h1>
        <h1> Student Age: {ogStudent.age} </h1>

       <input value={ogStudent.name} onChange={(e)=>
dispatch({type:"changeName", name: e.target.value})}/>
        <button onClick={()=> {dispatch({type: 'increment
age'})}}> Increase Age </button>
        <button onClick={()=> {dispatch({type: 'decrement
age'})}}> Decrement Age</button>
    </div>
  )
}
```

15. Write a program using useReducer that changes the color of the sentence on the basis of the fruit name that is chosen.

```
import React from 'react';
import { useReducer} from 'react';
```

```
const initFruit={
  name:"apple",
  theme:{color: "red"}
}

function changeFruit(state, action){
  switch(action.type){
    case "orange": return {...state, name: "orange",
theme:{color: "orange"}};
    case "mango": return {...state, name: "mango",
theme:{color:"yellow"}};
  }
}
export default function App(){
  let [fruit, dispatch]=useReducer(changeFruit, initFruit)
  return(
  <div>
    <h1 style={fruit.theme}>Pooja purchases to kg of
{fruit.name}</h1>

    <button onClick={()=>
dispatch({type:"orange"})}>Orange</button>
    <button onClick={()=>
dispatch({type:"mango"})}>Mango</button>
  </div>)
}
```

16. Write a program that demonstrates the rendering counts using useEffect, with no dependency array

```
import { useState, useEffect } from "react";
```

```
function Timer() {
  const [count, setCount] = useState(0);
  const [msg, setMsg] = useState('');
  function increaseCount(){
    setCount((prev) => prev + 1);
  }
  function changeMsg(event){
     setMsg(event.target.value);
  }
  useEffect(() => {
      console.log(count);
      console.log(msg);
  });

return (
  <div>
      <h1>I've rendered {count} times!</h1>
      <input type="text" placeholder="Enter the Message"
onChange={changeMsg}/>
      <button
onClick={()=>{increaseCount()}}>increaseCount</button>
  </div>
);
}
export default Timer;
```

17. Write a program that demonstrates the rendering counts using useEffect, using an empty dependency array.

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

```
function Timer() {
  const [count, setCount] = useState(0);
  const [msg, setMsg] = useState('');
  function increaseCount(){
   // window.alert("Hi");
    setCount((prev) => prev + 1);
  }
  function changeMsg(event){
     setMsg(event.target.value);
  }

  useEffect(() => {
      window.alert(count);
  },[]);

  return (
    <div>
        <h1>I've rendered {count} times!</h1>
        <input type="text" placeholder="Enter the Message"
onChange={changeMsg}/>
        <button
onClick={()=>{increaseCount()}}>increaseCount</button>
    </div>
  );
}
export default Timer;
```

Routing Programs

18. Write a program that demonstrates routing from one page to another.

App.js

```
import React from 'react';

import { BrowserRouter as Router, Routes, Route, useNavigate }
from 'react-router-dom';

import About from './About';


function GoToNextPage() {

  const navigate = useNavigate();


  const handleButtonClick = () => {

    navigate('/about');

  };


  return (

    <div>

      <h1>App Component</h1>

      <button onClick={handleButtonClick}>Go to About
Page</button>

    </div>

  );

}


export default function App() {

  return (

    <Router>

      <Routes>

        <Route path="/" element={<GoToNextPage />} />
```

```
        <Route path="/about" element={<About />} />
      </Routes>
    </Router>
  );
}
```

About.js

```
import React from 'react';

const About = () => {
  return (
    <div>
      <h1>About Page</h1>
    </div>
  );
};

export default About;
```

19. Write a program that demonstrates the use of parameters in routing.

App.js

```
import React, { useState } from 'react';
import { BrowserRouter as Router, Routes, Route, useNavigate } from 'react-router-dom';
import About from './About';

function GoToNextPage() {
  const [inputValue, setInputValue] = useState('');
  const navigate = useNavigate();
```

```
  const handleButtonClick = () => {
    navigate(`/about?input=${encodeURIComponent(inputValue)}`)
;
  };

  return (
    <div>
      <h1>App Component</h1>
      <input
        type="text"
        value={inputValue}
        onChange={(e) => setInputValue(e.target.value)}
        placeholder="Enter something"
      />
      <button onClick={handleButtonClick}>Go to About
Page</button>
    </div>
  );
}

export default function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<GoToNextPage />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </Router>
  );
```

```
}
```

About.js

```
import React from 'react';
import { useLocation } from 'react-router-dom';

const About = () => {
  const location = useLocation();
  const params = new URLSearchParams(location.search);
  const input = params.get('input');

  return (
    <div>
      <h1>About Page</h1>
      {input && <p>You entered: {input}</p>}
    </div>
  );
};
export default About;
```

20. Write a program to demonstrate nested routing.

App.js

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, useNavigate }
from 'react-router-dom';
import About from './About';
import Team from './Team';
```

```
function GoToNextPage() {
  const navigate = useNavigate();

  const handleButtonClick = () => {
    navigate('/about');
  };

  return (
    <div>
      <h1>App Component</h1>
      <button onClick={handleButtonClick}>Go to About
Page</button>
    </div>
  );
}

export default function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<GoToNextPage />} />
        <Route path="/about/*" element={<About />} />
      </Routes>
    </Router>
  );
}
```

About.js

```
import React from 'react';
```

```
import { Routes, Route, Link } from 'react-router-dom';
import Team from './Team';

const About = () => {
  return (
    <div>
      <h1>About Page</h1>
      <nav>
        <ul>
          <li>
            <Link to="team">Meet the Team</Link>
          </li>
        </ul>
      </nav>
      <Routes>
        <Route path="team" element={<Team />} />
      </Routes>
    </div>
  );
};

export default About;
```

Team.js

```
import React from 'react';

const Team = () => {
  return (
```

```
    <div>
      <h2>Team Page</h2>
      <p>Meet our awesome team!</p>
    </div>
  );
};

export default Team;
```