**Module – 3 ReactJs**

• What is React Js?

Ans:- React is declarative, efficient and flexible JAVASCRIPT library for building user interface.

• What is NPM in React Js?

Ans:- NPM is short for node package manager, an online directory that contains the various already registered open-source packages. NPM modules consume the various functions as a third-party package when installed into an app using the NPM command npm install.

• What is the Role of Node Js in react Js?

Ans:- You can definitely use [Node JS with React](https://programmers.io/reactjs-with-nodejs-development/) framework. In fact, Node JS is known to be the most suitable platform for hosting and running web servers for applications built on React. The two primary reasons for this are.

• What is CLI command In React Js?

Ans:- a command-line interface is a way that you can interact with a computer through text. It works by you typing in special commands in a command prompt.

• What are Components in React Js?

Ans:- Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

Components come in two types, Class components, and Function components, in this tutorial we will concentrate on Function components.

• What are Header and Content Components in React Js?

Ans:-

• How to install React Js on Windows, Linux Operating systems? How to install NPM and How to check the version of NPM?

Ans:-

**Step 1:**

Install NodeJS and NPM.

**Step 2:**

Install React.

**Command:**npm install -g create-react-app

**Step 3:**

Create a new React project.

**Command:** create-react-app project\_name

**Step 4:**

Combine the two steps in a single command.

**Command:**npx create-react-app project\_name

**Step 5:**

To get started, open the src folder and make changes in your desired file.

**Step 6:**

Start the server after the completion of the installation process.

**Command:**

* cd project\_name
* npm start

**Step 7:**

The port number will be displayed. Open the port in the browser.

• How to check the version of React Js?

Ans:- By typing ”npm -v” in cmd.

• How to change in components of React Js?

Ans:-

• How to Create a ListView in React Js?

Ans:-

• Create Increment decrement state change by button click?

CODE:

import React, { useState } from 'react';

function Setcount(props) {

    const [count,setcount] = useState(0)

    const [name,setname] = useState("yash")

    const handlestate =()=>{

        setname("raj")

    }

    return (

        <div>

            <h1>Count : {count}</h1>

            <h1>Name : {name}</h1>

            <button onClick={(e)=>setcount(count+1)}>Increment</button>

            <button onClick={(e)=>setcount(count-1)}>Decrement</button>

            <button onClick={(e)=>handlestate()}>Update</button>

        </div>

    );

}

export default Setcount;

Module – 4 Lists and Hooks

Q.1 Explain the Life cycle in Class Component and functional component with Hooks.

Ans:-

LifeCycle of ClassComponent:

import React, { Component } from 'react';

class Lifecycle extends Component {

    constructor(){

        super()

        this.state ={

            count:0

        }

        console.log('comp init')

    }

    handel=()=>{

        this.setState({

            count:10

        })

    }

    componentDidMount(){

        console.log("comp didmount")

    }

    render() {

        console.log('comp render')

        return (

            <div>

                {this.count}

                <button onClick={this.handel}>Click</button>

            </div>

        );

    }

    componentDidUpdate(){

        console.log('comp updated')

    }

}

export default Lifecycle;

->Function Component of Hooks

1. useState:

import React, { useState } from 'react';

function Setcount(props) {

    const [count,setcount] = useState(0)

    const [name,setname] = useState("yash")

    const handlestate =()=>{

        setname("raj")

    }

    return (

        <div>

            <h1>Count : {count}</h1>

            <h1>Name : {name}</h1>

            <button onClick={(e)=>setcount(count+1)}>Increment</button>

            <button onClick={(e)=>setcount(count-1)}>Decrement</button>

            <button onClick={(e)=>handlestate()}>Update</button>

        </div>

    );

}

export default Setcount;

2.useRef

import React, { useRef } from 'react';

function Useref(props) {

    const inputDOM = useRef()

    const handle =()=>{

        inputDOM.current.style.backgroundColor="red";

        inputDOM.current.id="fname"

        inputDOM.current.className="yash"

    }

    return (

        <div>

            <input type="text" ref={inputDOM} />

            <button onClick={handle}>Click</button>

        </div>

    );

}

export default Useref;

3.useMemo

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

function Usememo(props) {

    const [one,setone]=useState(0)

    const [two,settwo]=useState(0)

    const iseven = useMemo(()=>{

        for(var i=0;i<1000000000;i++){}

        if(one%2===0){

            return"even";

        }

    },[one])

    return (

        <div>

            <button onClick={()=>setone(one+1)}>{one}</button>

            {iseven?'even':'odd'}

            <button onClick={()=>settwo(two+1)}c>{two}</button>

        </div>

    );

}

export default Usememo;

4.useContext

import React, { createContext } from 'react';

import B from './B';

const name ={"name":"raj"}

const Firstname = createContext("")

function A(props) {

    return (

        <div>

            <h1>A</h1>

            <Firstname.Provider value={name}>

                <B />

            </Firstname.Provider>

        </div>

    );

}

export default A;

export {Firstname}

import React, { useContext } from 'react';

import { Firstname } from './A';

function D(props) {

    const firstname = useContext(Firstname)

    return (

        <div>

            <h1>D</h1>

            <h1>{firstname.name}</h1>

        </div>

    );

}

export default D;

5.customHook

import React from 'react';

function Fetch(x,y) {

    const sum = x+y

    return sum

}

export default Fetch;

import React, { useState } from 'react';

import Fetch from './Fetch';

function Callhook(props) {

    const data = Fetch(10, 20)

    const [num, setnum] = useState(0)

    const handleadd = () => {

        setnum(data)

    }

    return (

        <div>

            <h1>Add: {num}</h1>

            <button onClick={handleadd}>Add</button>

        </div>

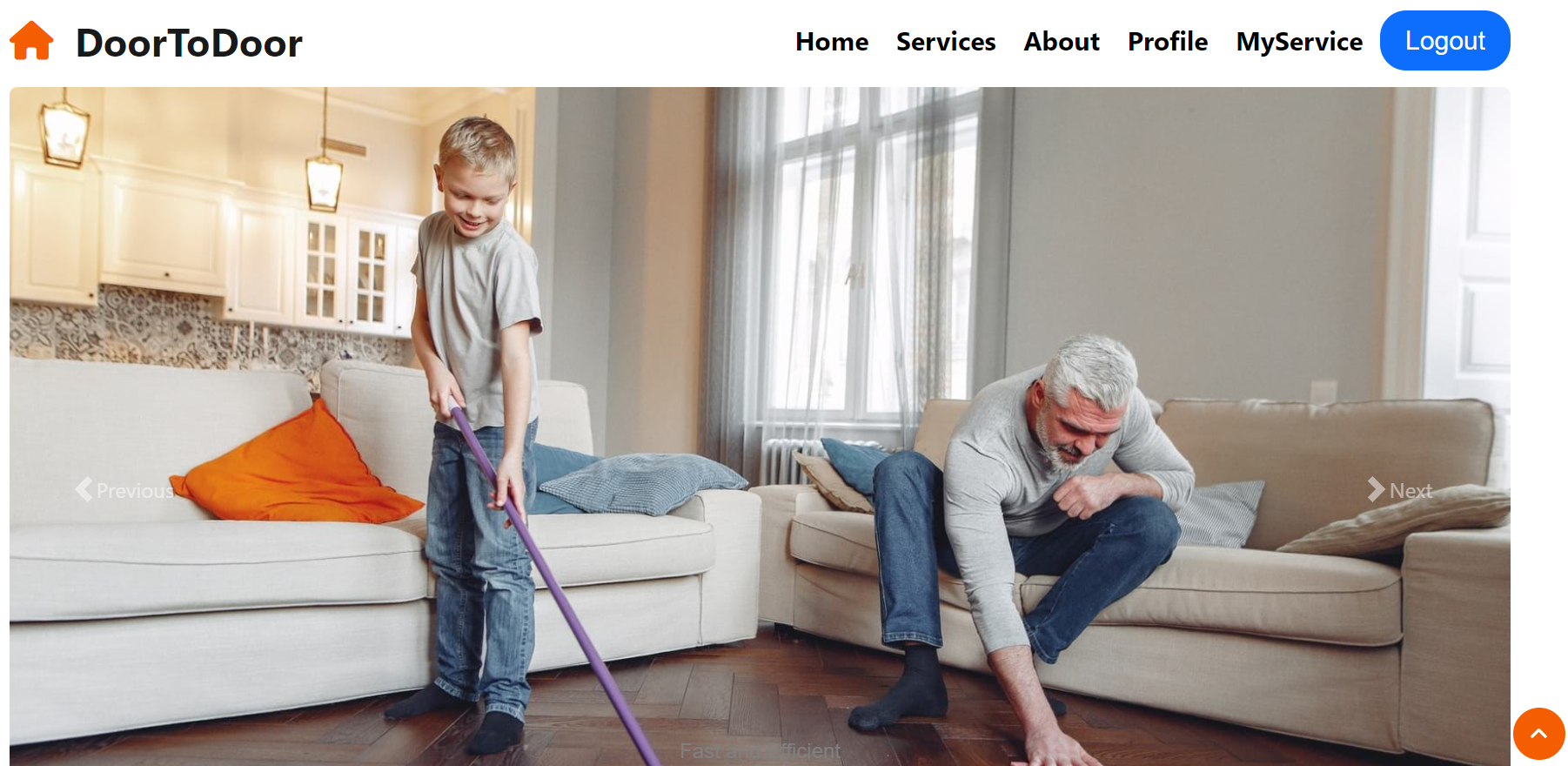
    );

}

export default Callhook;

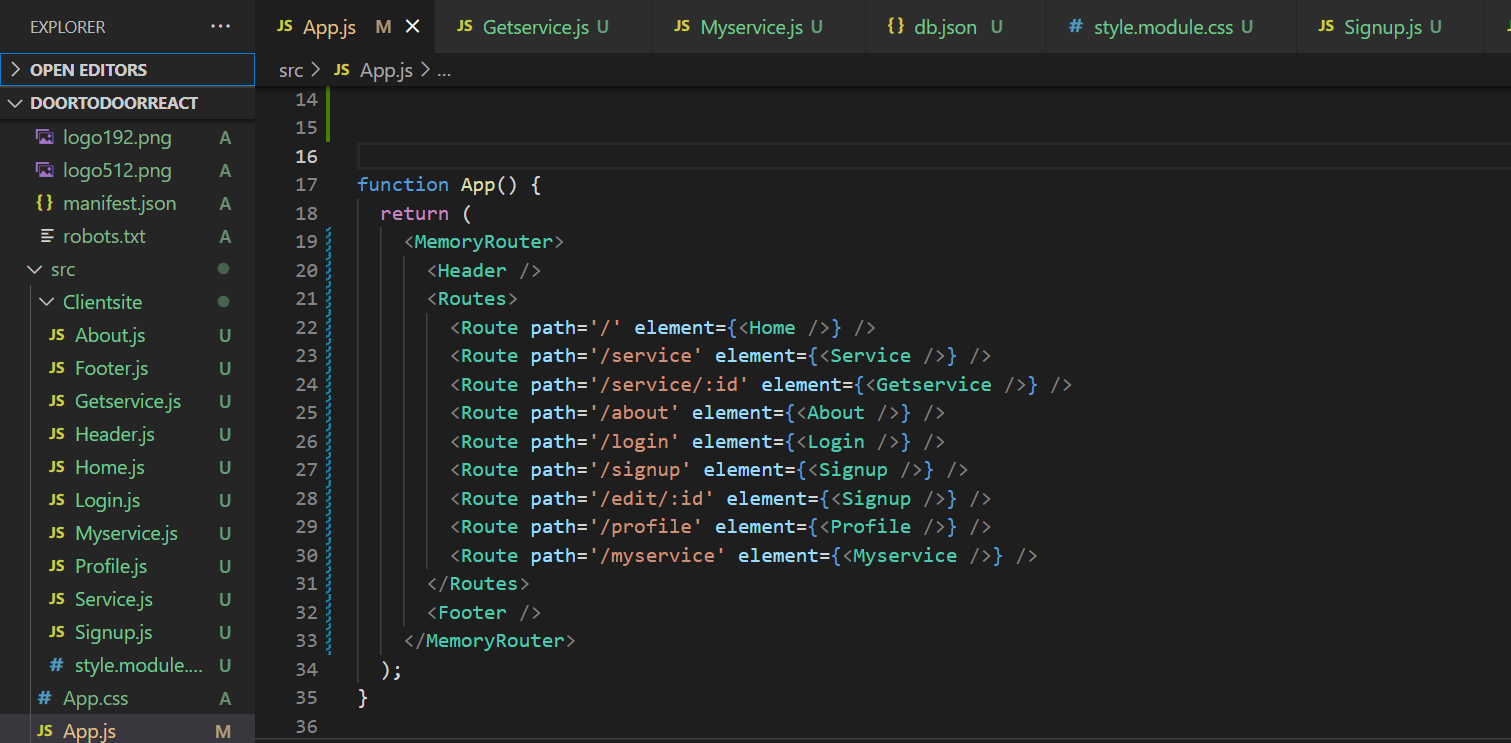
Module-5) React - Styling & Advance React

Q. Create a Shopping site home page with Styled- components



Module 6) React Router

• Create React app with modules and lazy loading (Admin-user module with child router and outlet)



Module 7) React- applying redux

• What is Redux?

Ans:- “Redux is a predictable state container for JavaScript apps”.

• What is Redux Thunk used for?

Ans:- Redux Thunk is a middleware that lets you call action creators that return a function instead of an action object. That function receives the store’s dispatch method, which is then used to dispatch regular synchronous actions inside the function’s body once the asynchronous operations have been completed.

• What is a Pure Component? When to use Pure Component over Component?

Ans:- PureComponent is exactly the same as Component except that it handles the shouldComponentUpdate method for you.

When props or state changes, PureComponent will do a ***shallow comparison*** on both props and state. Component on the other hand won’t compare current props and states to the next out of the box.

• What is the second argument that can optionally be passed to setState and what is its purpose?

Ans:- The second argument that can optionally be passed to setState is a callback function that gets called immediately after the setState is completed and the components get re-rendered.

• Create a Table and Search data from a table using React Js?

Ans:-

import React from 'react';

function Search(props) {

    const [search, setSearch] = React.useState('');

    const handleSearch = (event) => {

        setSearch(event.target.value);

    };

    const data = {

        nodes: list.filter((item) =>

            item.name.toLowerCase().includes(search.toLowerCase())

        ),

    };

    return (

        <div>

            <>

                <label htmlFor="search">

                    Search by Task:

                    <input id="search" type="text" onChange={handleSearch} />

                </label>

                <Table data={data}>

                    ...

                </Table>

            </>

        </div>

    );

}

export default Search;