**Advance jS crude**

**Html file**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">

    <link rel="stylesheet" href="styles/style.css">

</head>

<body>

    <div class="container-fluid py-2 header position-sticky top-0">

        <div class="container">

            <div class="nav">

                <div id="left" class="px-3">

                    <h2 class="">STUDENT</h2>

                </div><span href="" data-bs-toggle="modal" data-bs-target="#staticBackdrop"></span>

                <div id="right" class="ms-auto">

                    <button class="btn mt-1 hbtn" data-bs-toggle="modal" data-bs-target="#staticBackdrop">+ADD</button>

                </div>

            </div>

        </div>

    </div>

    <!-- Modal -->

    <div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1"

        aria-labelledby="staticBackdropLabel" aria-hidden="true">

        <div class="modal-dialog">

            <div class="modal-content">

                <div class="modal-header">

                    <h4 class="modal-title">My Personal Information</h4>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"

                        onclick="dismis()"></button>

                </div>

                <div class="modal-body">

                    <form id="form" onsubmit="createFn(event)">

                        <div class="row row-cols-1 row-cols-md-2">

                            <div class="col">

                                <div class="mb-3">

                                    <label for="inputOne" class="form-label">First Name</label>

                                    <input type="text" class="form-control" id="inputOne" placeholder="First name">

                                </div>

                                <div class="mb-3">

                                    <label for="inputTwo" class="form-label">Last Name</label>

                                    <input type="text" class="form-control" id="inputTwo" placeholder="Last name">

                                </div>

                                <div class="mb-3">

                                    <label for="inputThree" class="form-label">Email address</label>

                                    <input type="email" class="form-control" id="inputThree" placeholder="Email">

                                </div>

                                <div>

                                    <label for="hobbyDiv">Hobby</label>

                                    <div id="hobbyDiv" class="ps-2">

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input check" type="checkbox" value="cricket"

                                                id="checkOne" name="box">

                                            <label class="form-check-label" for="flexCheckDefault">Cricket</label>

                                        </div>

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input check" type="checkbox" value="football"

                                                id="checkTwo" name="box">

                                            <label class="form-check-label" for="flexCheckDefault">Football</label>

                                        </div>

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input check" type="checkbox" value="volleyball"

                                                id="checkThree" name="box">

                                            <label class="form-check-label" for="flexCheckDefault">Volleyball</label>

                                        </div>

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input check" type="checkbox" value="carom"

                                                id="checkFour" name="box">

                                            <label class="form-check-label" for="flexCheckDefault">Carom</label>

                                        </div>

                                    </div>

                                </div>

                            </div>

                            <div class="col">

                                <div class="mb-3">

                                    <label for="inputFour" class="form-label">Age</label>

                                    <input type="number" class="form-control" id="inputFour" placeholder="Your Age">

                                </div>

                                <div class="mb-3">

                                    <label for="inputFive" class="form-label">Phone No</label>

                                    <input type="tel" class="form-control" id="inputFive" placeholder="Your Mobile No">

                                </div>

                                <div class="mt-2">

                                    <label for="radioDiv">Gender</label>

                                    <div id="radioDiv" class="ps-2">

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input radio" type="radio" value="male"

                                                id="radioOne" name="gender">

                                            <label class="form-check-label" for="flexCheckDefault">Male</label>

                                        </div>

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input radio" type="radio" value="female"

                                                id="radioTwo" name="gender">

                                            <label class="form-check-label" for="flexCheckDefault">Female</label>

                                        </div>

                                        <div class="form-check form-check-inline">

                                            <input class="form-check-input radio" type="radio" value="trans"

                                                id="radioThree" name="gender">

                                            <label class="form-check-label" for="flexCheckDefault">Trans</label>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <div class="modal-footer mt-2">

                            <button type="submit" class="btn btn-success" data-bs-dismiss="modal"

                                id="submitBtn">Submit</button>

                            <button type="button" class="btn btn-danger" data-bs-dismiss="modal" id="cancelBtn"

                                onclick="dismis()">Cancel</button>

                            <button type="reset" class="btn btn-secondary" id="resetBtn">Reset</button>

                        </div>

                    </form>

                </div>

            </div>

        </div>

    </div>

    <div class="container-fluid mt-5 rounded-top-3">

        <div class="g-0 rounded-top-3">

            <div class="table-responsive rounded-top-3" id="tablenew">

                <table class="overflow-auto" id="table">

                    <caption class="fw-bolder text-center caption-top text-uppercase">Student personal information

                    </caption>

                    <thead id="thead" class="text-bg-danger">

                        <tr class="">

                            <th>Id</th>

                            <th>First name</th>

                            <th>Last name</th>

                            <th>Email</th>

                            <th>Age</th>

                            <th>Telephone</th>

                            <th>Gender</th>

                            <th>Hobby</th>

                            <th>Action</th>

                        </tr>

                    </thead>

                    <tbody id="tbody">

                    </tbody>

                </table>

            </div>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"

        crossorigin="anonymous"></script>

    <script src="js/script.js"></script>

</body>

</html>

**JS File**

/\* =============Creat of Crud================ \*/

const form = document = document.getElementById("form");

const table = document.getElementById("tbody");

let mainArray = [];

let editIndex;

let editId = null;

const setLocalStorage = () => {

    localStorage.setItem('mainArray', JSON.stringify(mainArray));

}

const getLocalStorage = () => {

    mainArray = JSON.parse(localStorage.getItem('mainArray'));

}

const createFn = (event) => {

    event.preventDefault();

    let firstName = document.getElementById("inputOne").value;

    let lastName = document.getElementById("inputTwo").value;

    let email = document.getElementById("inputThree").value;

    let age = document.getElementById("inputFour").value;

    let tel = document.getElementById("inputFive").value;

    let gender = document.querySelector('input[name="gender"]:checked').value;

    let hobby = Array.from(document.querySelectorAll('input[name="box"]:checked')).map((item) => {

        return item.value;

    });

    const mainObj = {

        id: editId ? editId : Date.now(),

        firstName,

        lastName,

        email,

        age,

        tel,

        gender,

        hobby

    }

    if (editId) {

        mainArray[editIndex] = mainObj;

        setLocalStorage();

        tableSet();

        editId = null;

        form.reset();

    }

    else {

        mainArray.push(mainObj);

        setLocalStorage();

        tableSet();

        form.reset();

    }

}

/\* =============Read of Crud================ \*/

const tableSet = () => {

    getLocalStorage();

    let rowData = '';

    mainArray.map((obj) => {

        rowData +=

            `<tr>

                <td class="${obj.id}">${obj.id}</td>

                <td>${obj.firstName}</td>

                <td>${obj.lastName}</td>

                <td>${obj.email}</td>

                <td>${obj.age}</td>

                <td>${obj.tel}</td>

                <td>${obj.gender}</td>

                <td>${obj.hobby}</td>

                <td><button class="edit btn" onclick="editFn(${obj.id})"><span data-bs-toggle="modal" data-bs-target="#staticBackdrop">Edit</span></button><button class="delete btn" onclick="deleteFn(${obj.id})">Delete</button></td>

            </tr>`

    })

    table.innerHTML = rowData;

}

window.addEventListener("load", () => {

    if ((localStorage.getItem("mainArray") !== null) && (localStorage.getItem("mainArray") !== null)) {

        tableSet();

    }

})

/\* =============== Delete Process ================== \*/

const deleteFn = (id) => {

    if (confirm("Do you want to delete this row?") === true) {

        mainArray.splice(mainArray.findIndex(x => x.id === id), 1)

        setLocalStorage();

        getLocalStorage();

        tableSet();

    }

}

/\* =============== Edit Process ==================== \*/

const editFn = (id) => {

    editId = id;

    editClick = true;

    let obj

    mainArray.forEach((item, index) => {

        if (item.id === id) {

            obj = item;

            editIndex = index;

        }

    })

    document.getElementById("inputOne").value = obj.firstName;

    document.getElementById("inputTwo").value = obj.lastName;

    document.getElementById("inputThree").value = obj.email;

    document.getElementById("inputFour").value = obj.age;

    document.getElementById("inputFive").value = obj.tel;

    (document.querySelector(`input[value="${obj.gender}"]`)).checked = true;

    obj.hobby.forEach((getItem) => {

        document.querySelector(`input[value="${getItem}"]`).checked = true;

    })

}

/\* ========================Dismis=>when we click cancel sign of form to remove form data============== \*/

const dismis = () => {

    form.reset();

}

**CSS File**

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    /\* border: 1px solid black; \*/

}

nav button {

    width: 100px;

}

/\* =========Table========= \*/

#table {

    width: 100%;

    text-align: center;

}

#tablenew{

    box-shadow:5px 5px 10px grey,-5px -5px 20px grey;

}

tr,

td,

th {

    border: 1px solid black;

}

#table caption {

    border-radius: 10px 10px 0 0;

    font-size: 25px;

    color:crimson;

}

#thead {

    font-size: 22px;

}

#tbody tr {

    color: black;

    font-size: 18px;

    font-weight: 600;

}

.edit, .delete {

    border-radius: 10px;

    color: wheat;

    box-shadow: 2px 2px 4px grey;

    width: 80px;

    margin: 0 5px;

    font-weight: 600;

}

.header {

    background-image: linear-gradient(to right, #434343 0%, black 100%);

}

.hbtn {

    font-size: 25px;

    color: #434343;

    font-weight: 600;

    font-family: serif;

}

#left {

    display: flex;

    align-items: center;

    border-radius: 30px;

}

#left h2 {

    margin: 0;

    padding: 0;

    color: white;

}

.header .btn {

    width: 150px;

    border-radius: 30px;

    color: black;

    background-color: aliceblue;

}

.header .btn:hover {

    box-shadow: inset -5px -5px 2px grey;

}

/\* form \*/

#form label {

    color: black;

    font-weight: 600;

}

#form input {

    font-weight: 500;

    font-family: Verdana, Geneva, Tahoma, sans-serif;

}

.edit span , .delete span{

    display: inline-block;

    width: 100%;

}

.edit , .delete{

    padding: 0;

    color: white;

    font-weight: 600;

    background-color: rgba(0, 0, 0, 0.355);

}

.edit{

    background-color: orange;

}

.delete{

    background-color: indianred;

}

.modal-dialog{

    max-width: 700px;

    border: 4px solid black;

    border-radius: 10px;

}

.btn{

    border: 1px solid black;

    margin: 5px 5px;

}

.bs-btn-border-radius{

    border-radius: 0;

}

**Process Of make React Project**

Step1=>Install node js

Step2=>Check version of node using command prompt =>  node -v

      if give version that means node succesfully installed

Step3=>To make new project open vs code and run terminal using shortcut key ctrl + ` or  from menu bar

Step4=>Then check we are in which directory.If we are out of folder in which we want to make code then enter in that directory using => cd foldername

then write =>npx create-react-app projectname

        or =>npx create-react-app .(fullstop sign)  =>jo aa rite run karyu hoy to je folder ma hoy a name no project a folder ma bani jase

==>Project name conditions =>all character in lowercase

                             no any space or special character except hyphen,undersore=> - \_

Step5=>after project create run that project using npm start command

**Use of Fragment**

/\* fragment is one type of div but not take space and not show in dom. it is use because we can return one item only from one function therefore we put that jsx formate HTML in that Fragment. it is act as one block or one item\*/

**Import Export in React**

we make component in react and then to use in another file we have to export that component.

We can export using two method

1=>export

2=>export default

1=>export write when that component define

Ex=>

export class FirstClassComponent extends Component {

    render() {

        return (

/\* fragment is one type of div but not take space and not show in dom. it is use because we can return one item only from one function therefore we put that jsx formate HTML in that Fragment. it is act as one block or one item\*/

            <Fragment>

                <h1>Hello</h1>

                <h2>Good Morning!</h2>

            </Fragment>

        )

    }

}

2=>export default we can write at two places when define or after define

  1=>

    export default class FirstClassComponent extends Component {

    render() {

        return (

            <>

                <h1>Hello</h1>

                <h2>Good Morning!</h2>

            </>

        )

    }

    }

  2=>

    import React, { Component, Fragment } from 'react'

    class FirstClassComponent extends Component {

    render() {

        return (

            <>

                <h1>Hello</h1>

                <h2>Good Morning!</h2>

            </>

        )

    }

  }

export default FirstClassComponent

difference of export & export default

1=>When only export use then during import we have to write same name which is using define and also put in curly bracket like below example

ex=>

    import { FunctionalComponent } from './components/FunctionalComponent'

              name which is we using define

ex=>

    import Anyname from './components/FunctionalComponent'

            we can change this name to any name but path is not change

How to select path=>

if we want to back from that file then use=> ./

if we want to back from that folder then use=> ../

**Functional Component**

export const FunctionalComponent = () => {

    return (

        <>

            <h1>Hello</h1>

            <h2>Good Morning!</h2>

        </>

    )

}

**Class Component**

import React, { Component, Fragment } from 'react'

class FirstClassComponent extends Component {

 /\*In class component we have to call render function compulsory \*/

    render() {

        return (

            <>

                <h1>Hello</h1>

                <h2>Good Morning!</h2>

            </>

        )

    }

}

export default FirstClassComponent

**App.js**

import './App.css'

import { FunctionalComponent } from './components/FunctionalComponent'

// import FunctionalComponent from './components/FunctionalComponent'

const App = () => {

  return (

    <div className="App">

      <FunctionalComponent />

    </div>

  )

}

export default App

**Import Export Main Point**

**First check karvu ke aakhi file import karvani che ke only function.**

**1)Jo only function import karvu hoy to or jo file ni under**

**1=>Jo export karyu hoy to**

import { FirstFunction } from './components/FirstFunction';

**2=>Jo export default karyu hoy to**

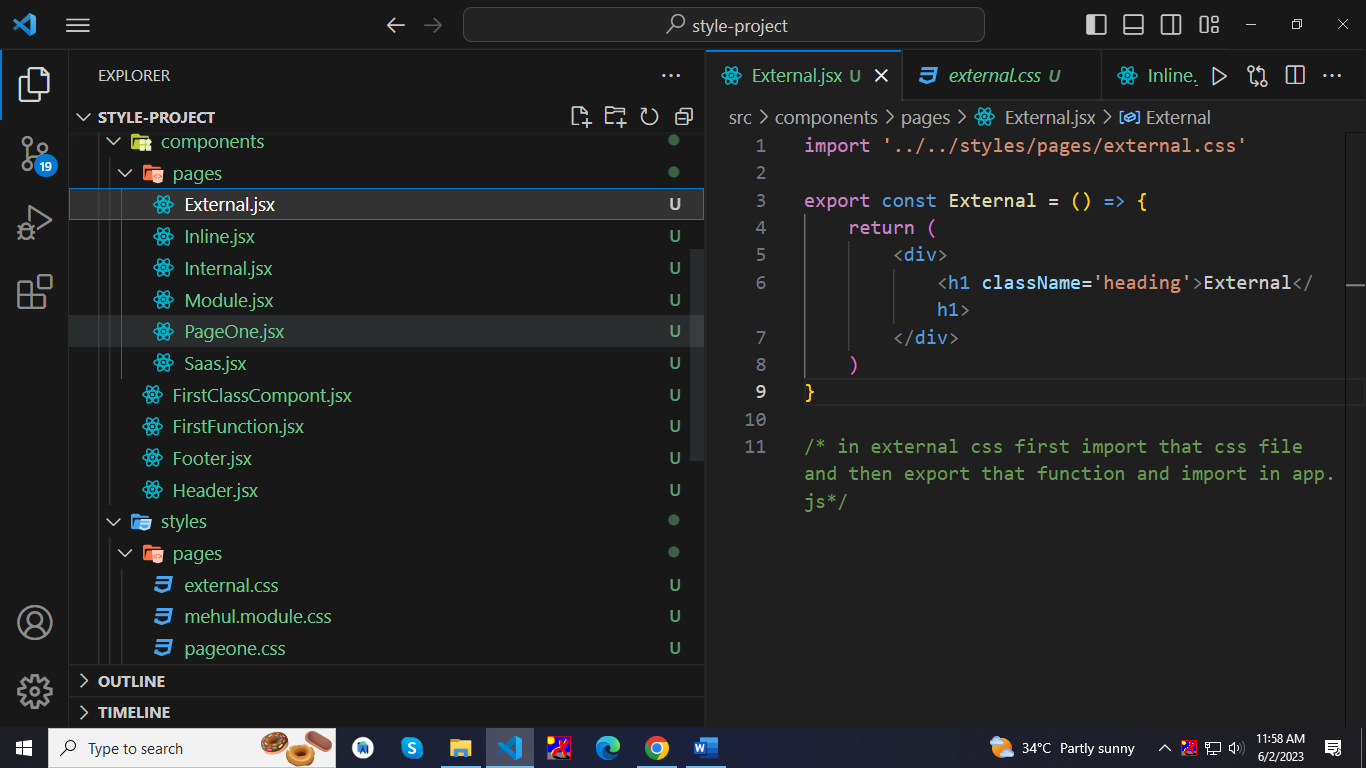
import FirstClassComponent from './components/FirstClassCompont';

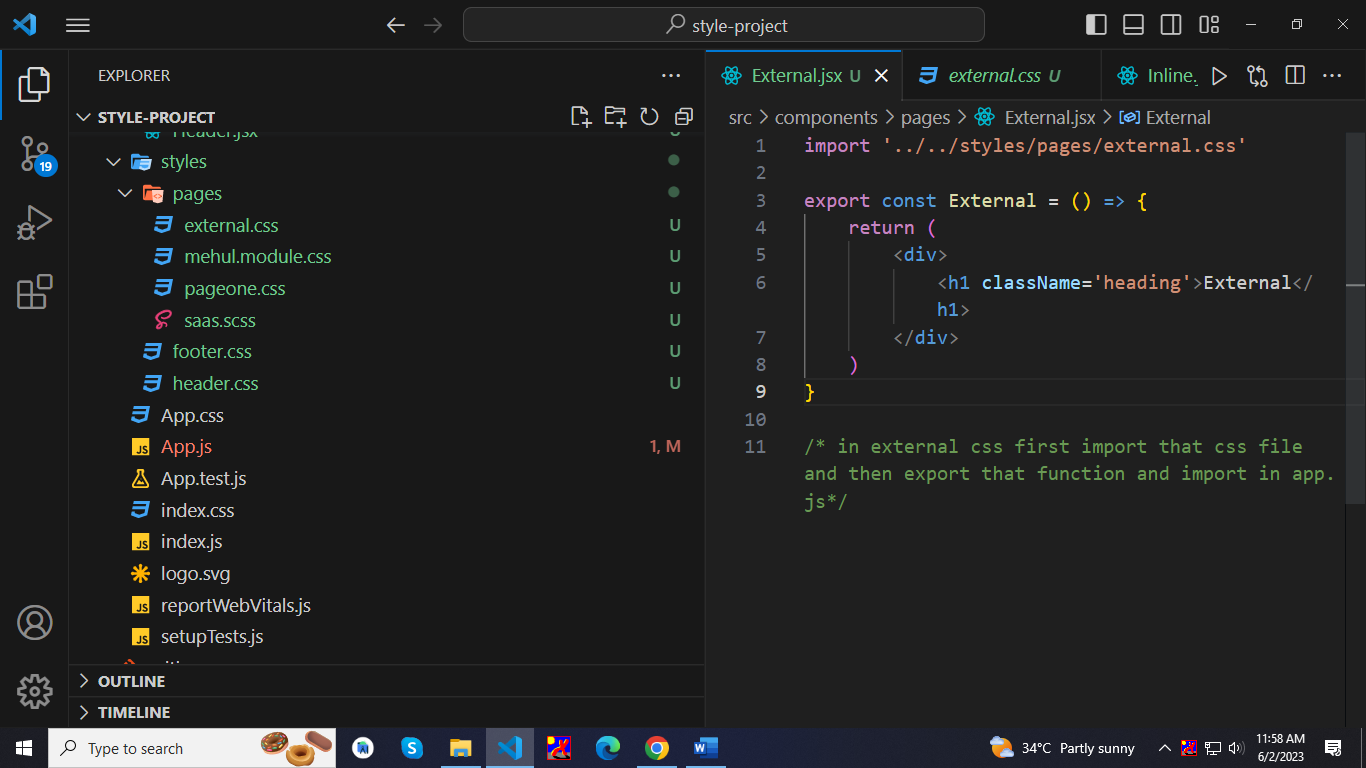
**ane aapne FirstClassComponent ni jagyae koi biju name pan rakhi shakiye.**

**2)Jo file Import karvi hoy to**

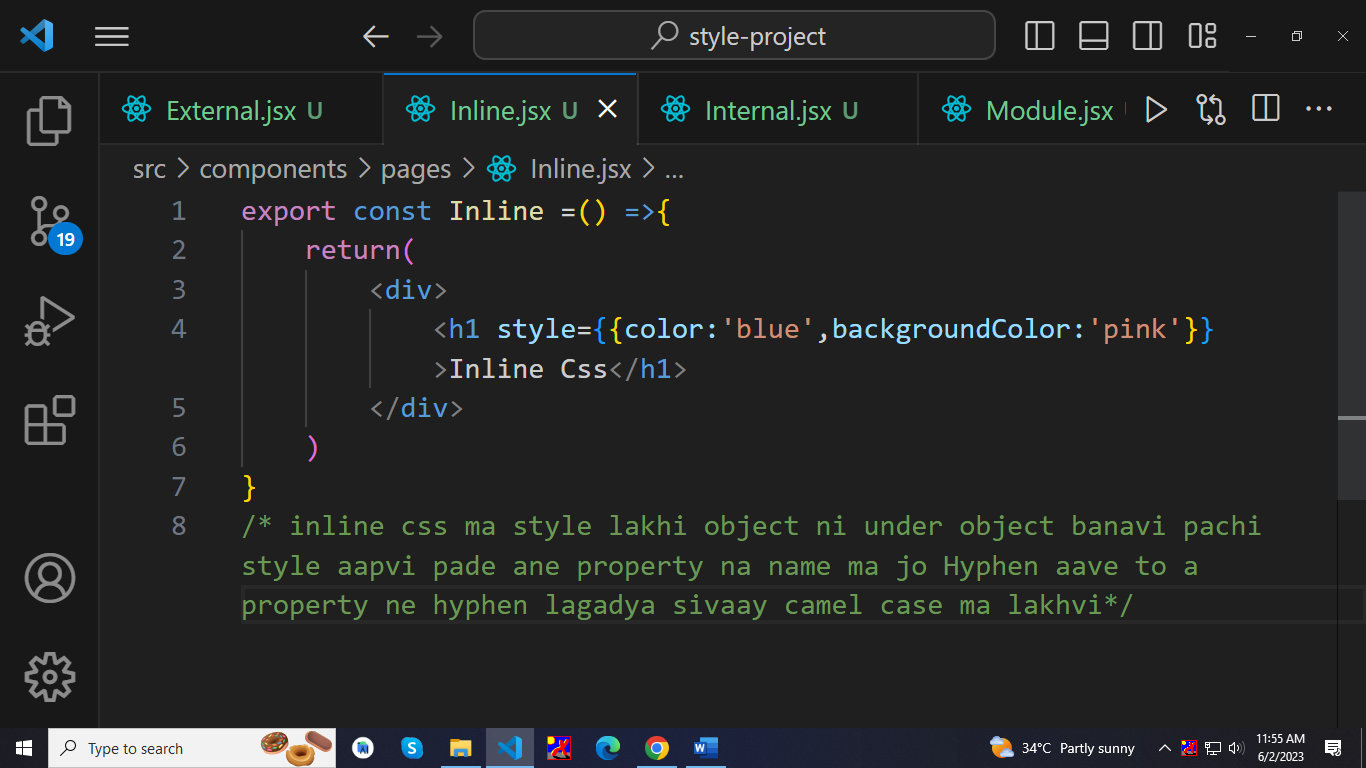
import './App.css';

Folder Structure

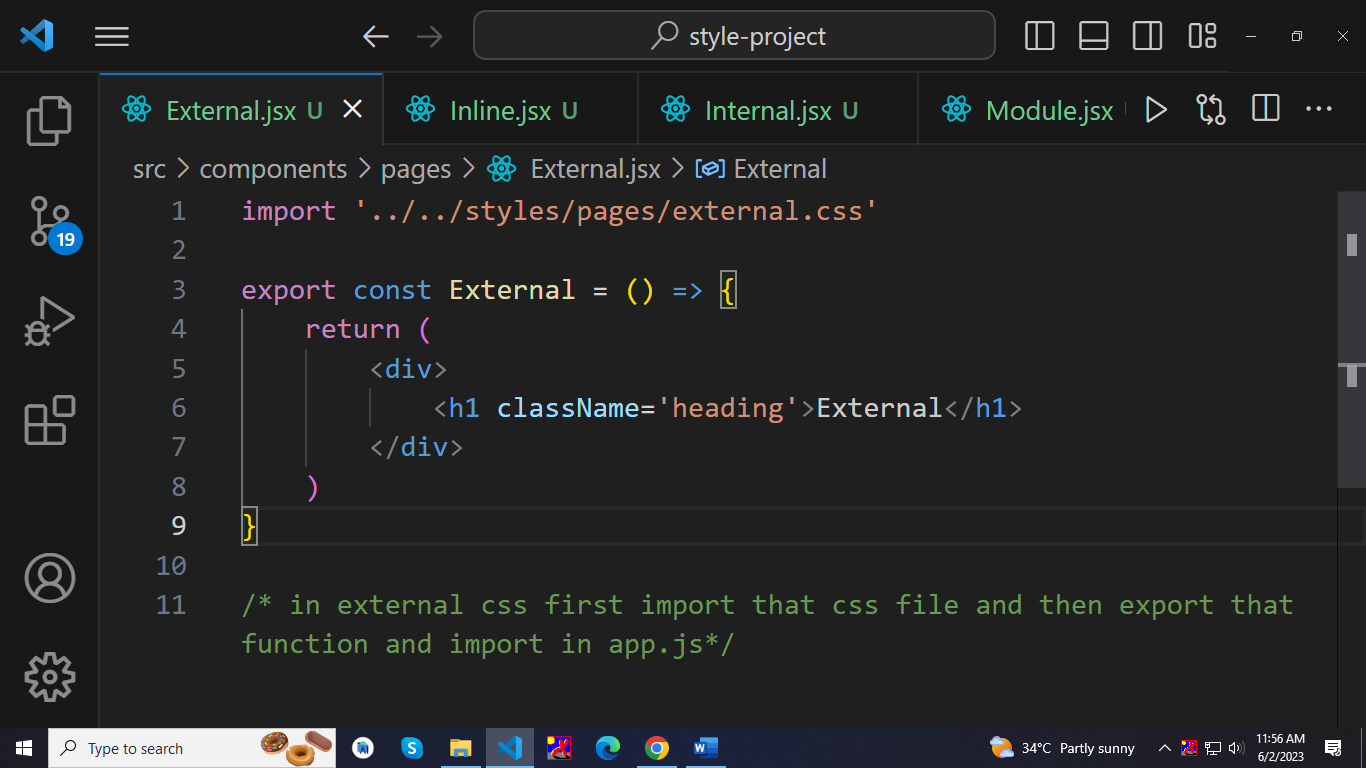


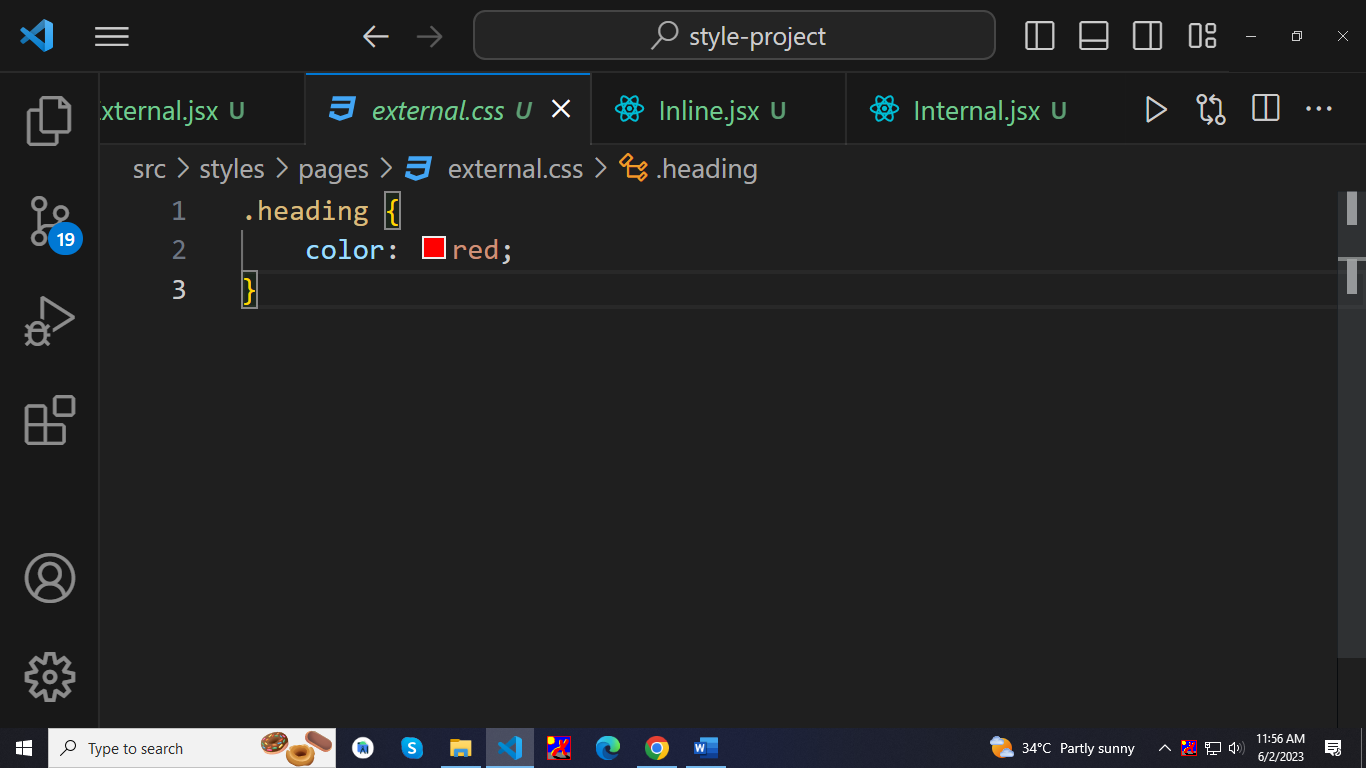


**Inline Style**

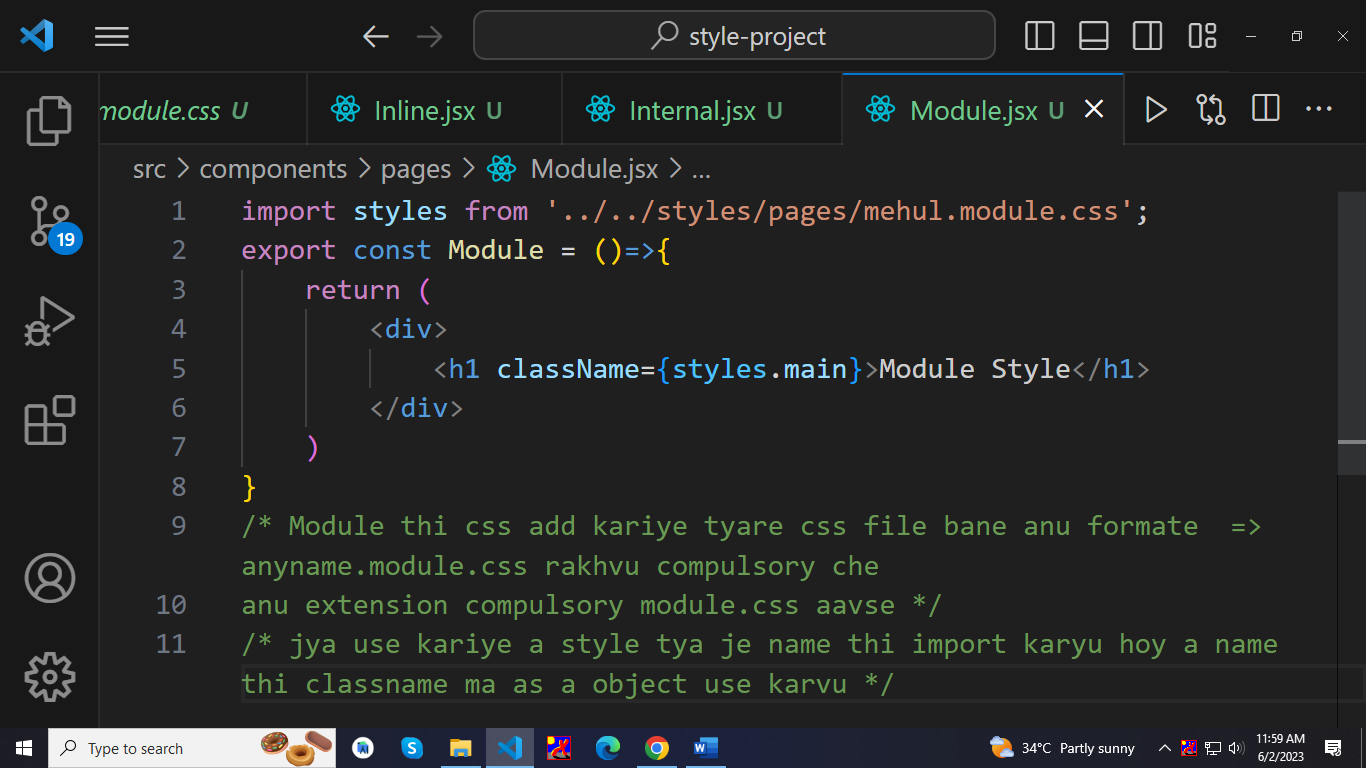


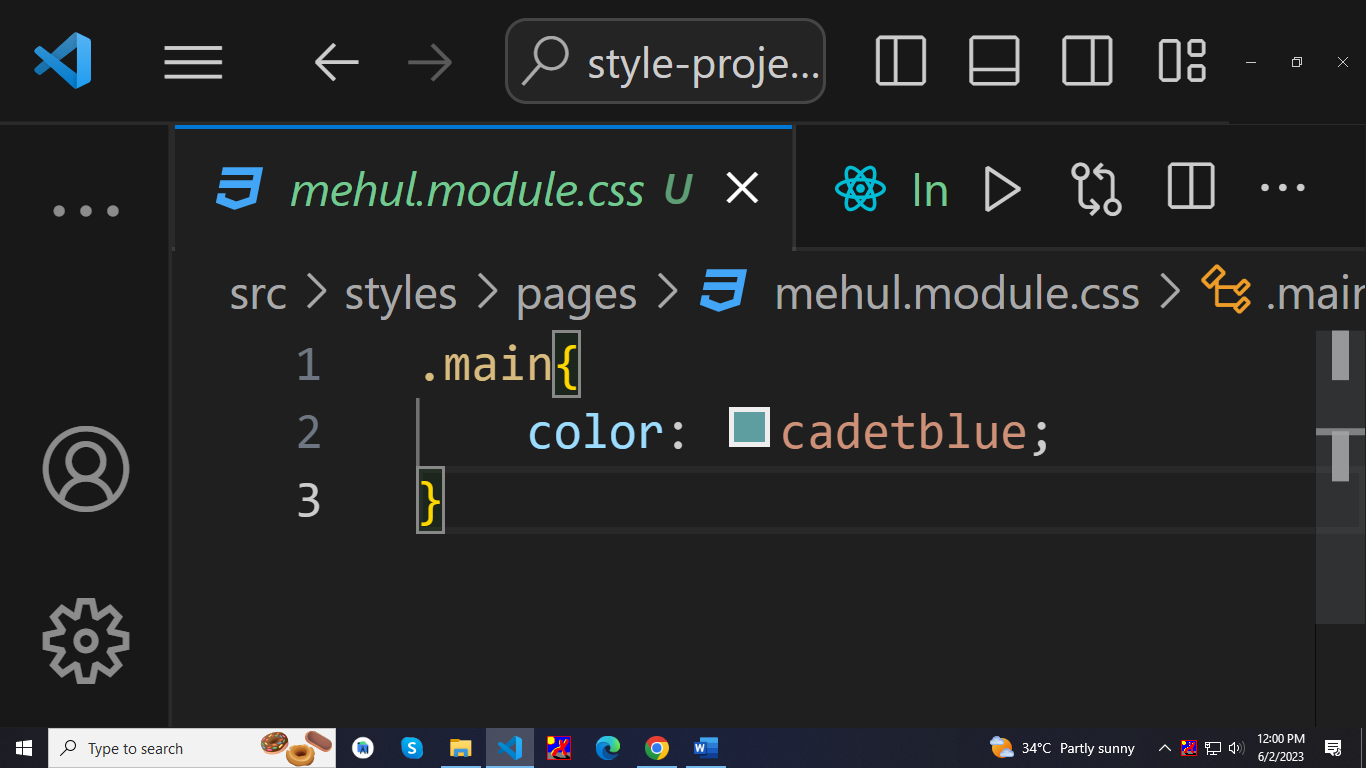
**External Style**



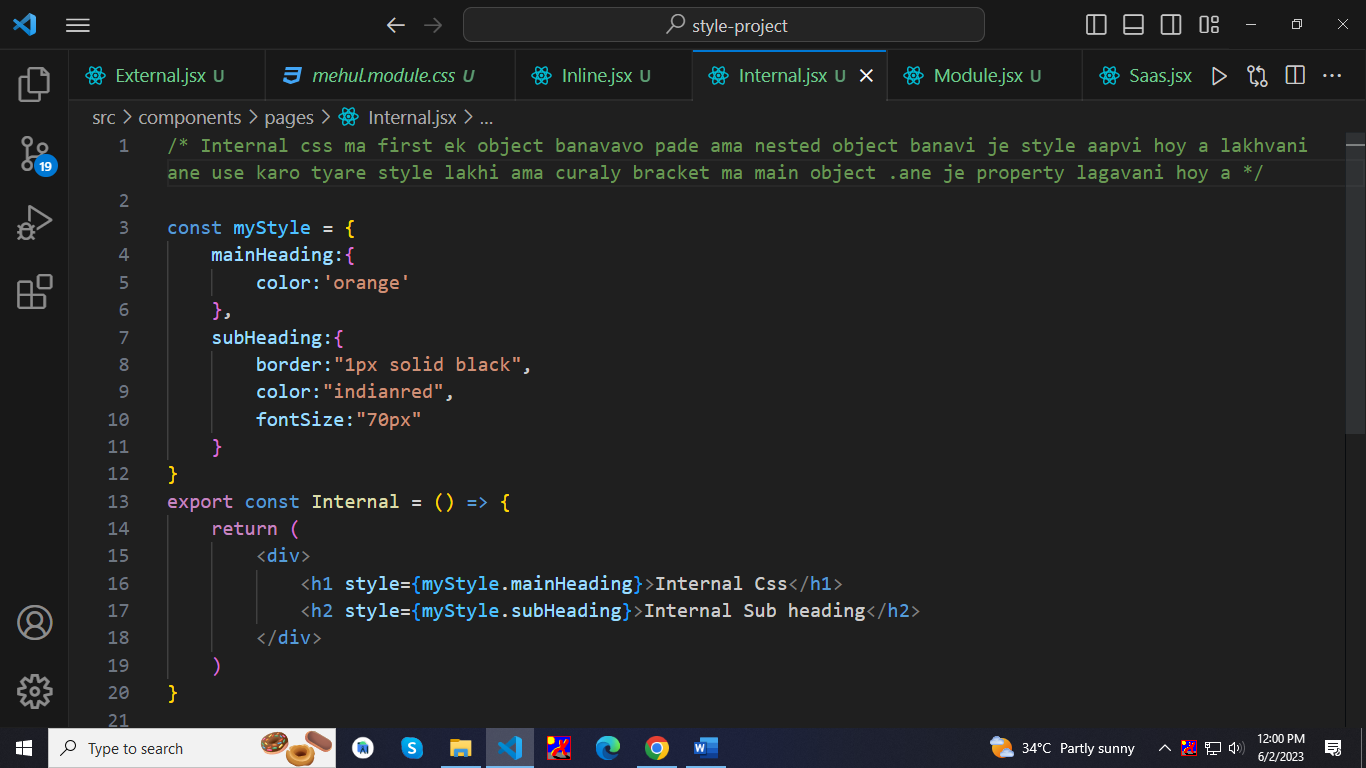


**Module CSS**

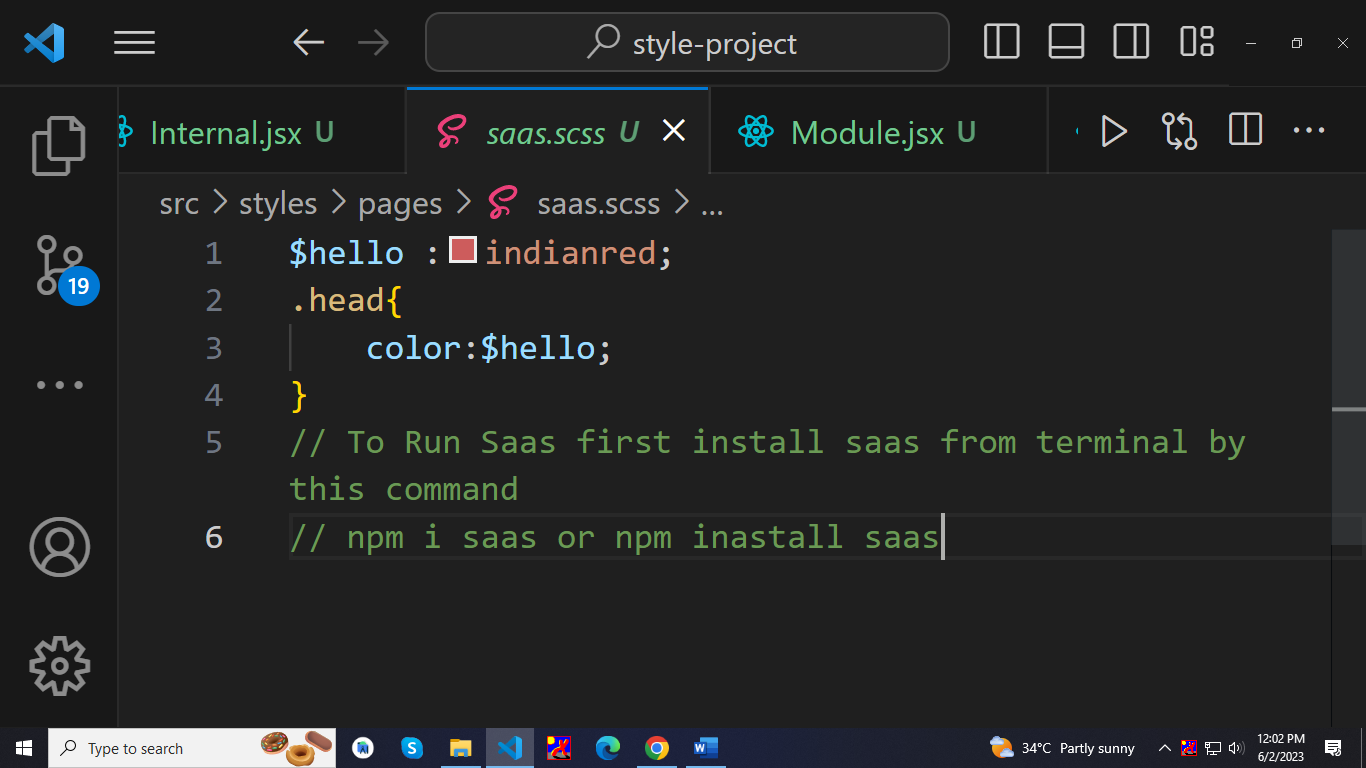


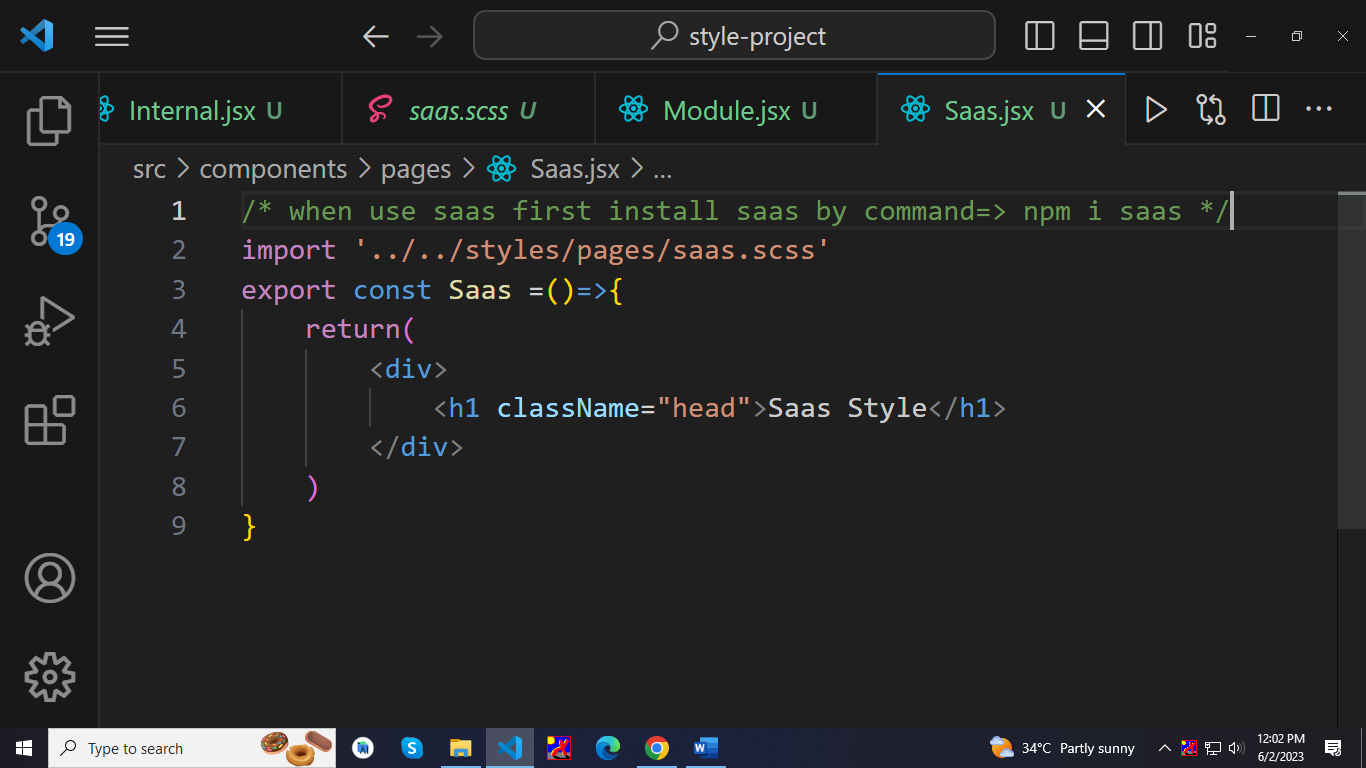


**Internal css**



**Saas CSS**





App

Js file

import './App.css';

import FirstClassComponent from './components/FirstClassCompont';

import { FirstFunction } from './components/FirstFunction';

import { External } from './components/pages/External';

import { Inline } from './components/pages/Inline';

import { Saas } from './components/pages/Saas';

import { Module } from './components/pages/Module';

import {Internal} from './components/pages/Internal';

function App() {

  return (

    <div className="App">

      <FirstClassComponent />

      <FirstFunction />

      <External />

      <Inline />

      <Saas />

      <Module />

      <Internal />

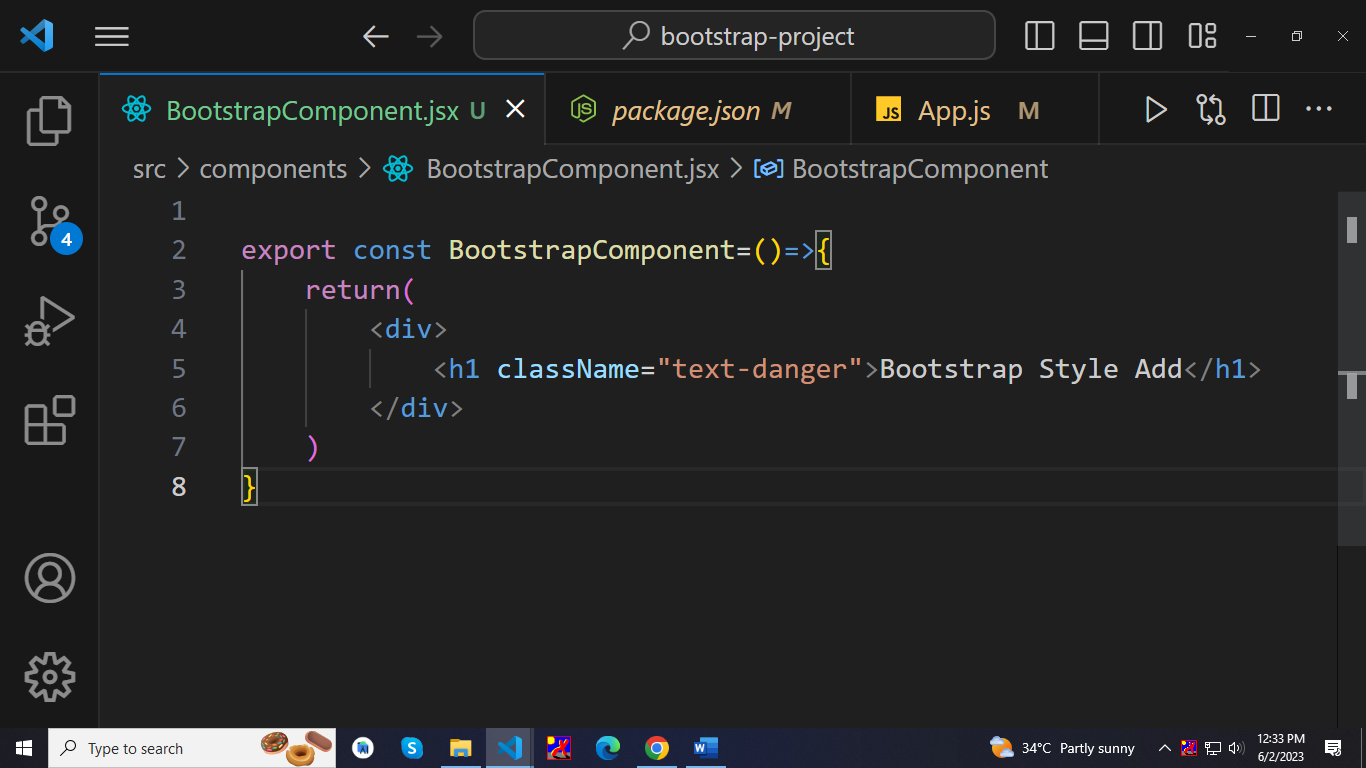
    </div>

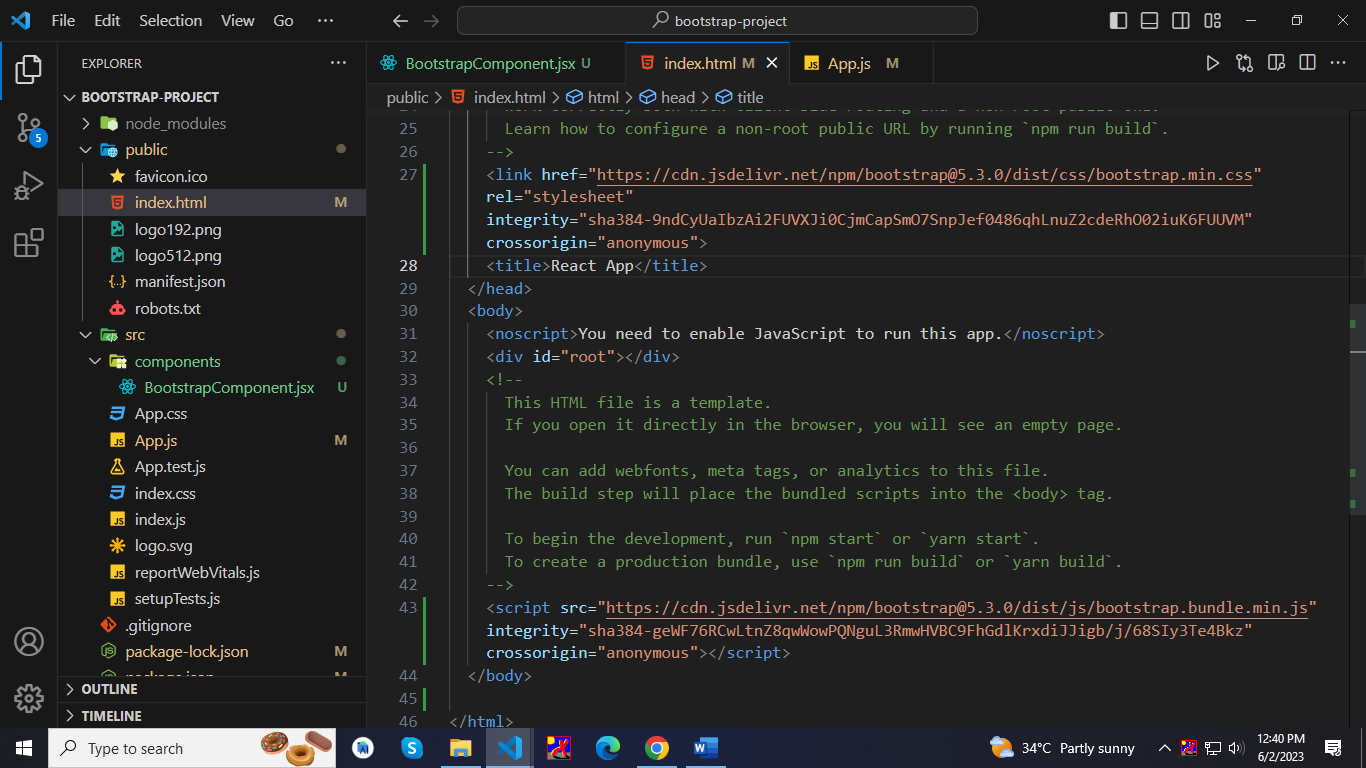
  );

}

export default App;

Bootstrap





In index.html file bootstrap css file and and js file link

And when we use bootstrap that time which tag is self close that tag we have close like this

<input />

And Class replace by className like this and self close also /

<input className="form-control me-2" type="search" />

Fragment use=>it is remove branch and all components are make in single div, not make branch of all div

**Class state**

State is variable .In java script it is calls variable and in react it is calls state.

In react We can not create variable or state directly.

Constructor is a natural space of class component which use for declare state.

It is make using constructor and it is use in that class component like this example.

**Update or Change State**

import React, { Component, Fragment } from 'react'

class StateComponent extends Component {

    constructor() {

        super()

        this.state = {

            fullName: 'Mehul Balvantbhai Waghela',

            phone: 6355173787,

            email: 'mehulwaghela63@gmail.com',

            hobbies: ['Dancing', 'Travelling', 'Coding'],

            address: {

                home: '71,Deepnagar society,kim',

                office: 'Oscar Career Point'

            },

            online: true

        }

    }

    /\* in class based function we can make function like this without let or const \*/

    ChangState=(a)=>{

        console.log(a);

        this.setState({online: !this.state.online})

    }

    render() {

        return (

            <Fragment>

                <h1>My Personal Information</h1>

                <p><strong>Full Name:-</strong>{this.state.fullName}</p>

                <p><strong>Phone:-</strong>{this.state.phone}</p>

                <p><strong>Email:-</strong> {this.state.email}</p>

                <p><strong>Hobbies:-</strong> {this.state.hobbies.join(', ')}</p>

                <p><strong>Home Address:-</strong> {this.state.address.home}</p>

                <p><strong>Office Address:- </strong>{this.state.address.office}</p>

                <p><strong>online:-</strong> {this.state.online? 'Yes':'NO'}</p>

                <button onClick={this.ChangState}>{(!this.state.online === true)? "online":"offline"}</button>

                {/\* if we want to send paramenter like this using anonympus function \*/}

                <button onClick={()=>{this.ChangState(1)}}>{(!this.state.online === true)? "online":"offline"}</button>

                {/\* If we write code like this than it is directly call function withput click therefore we can not put round braces after function name. if we have requirement of pass argument then use  onClick={()=>{this.ChangState(1)} & if we dont have requirement of pass argument than use onClick={this.ChangState}\*/}

                {

                /\* this is wrong =>because this call function continuously <button onClick={this.ChangState()}>{(!this.state.online === true)? "online":"offline"}</button> \*/}

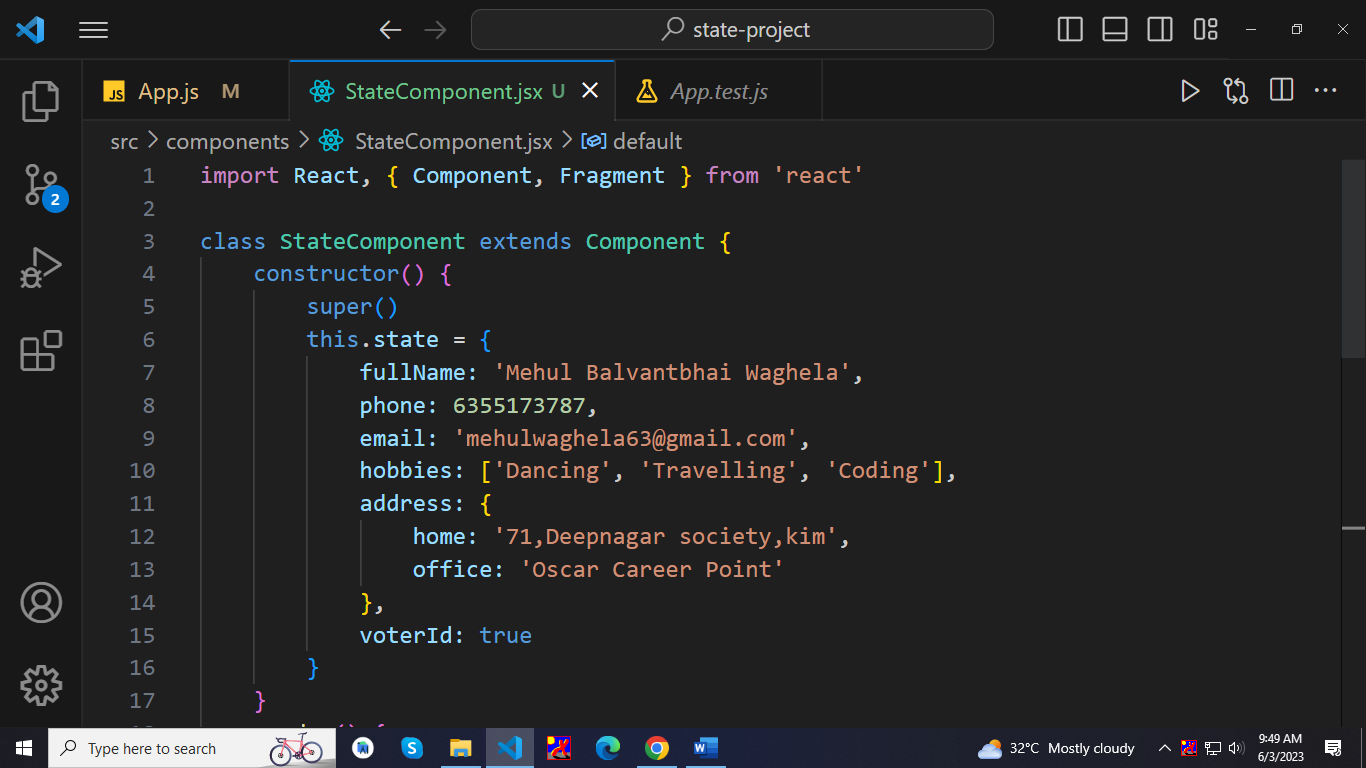
            </Fragment>

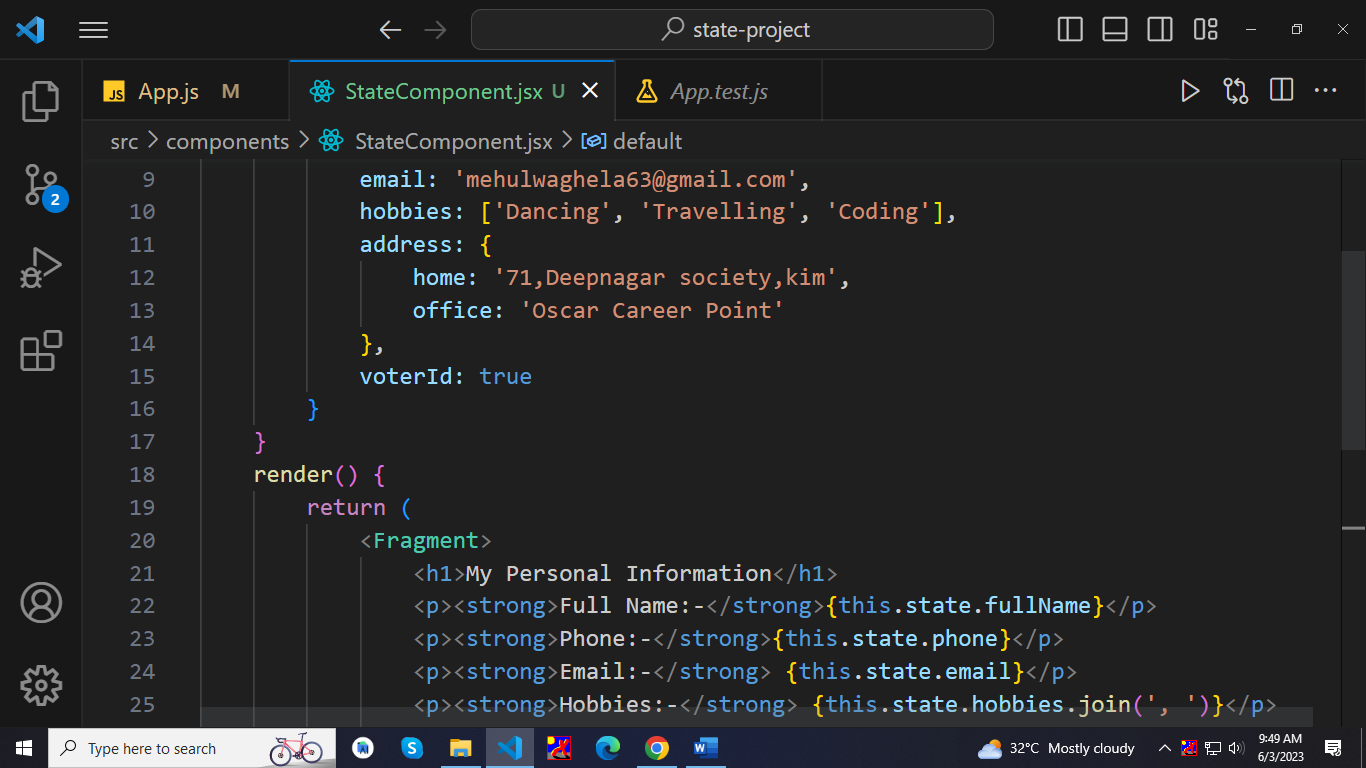
        )

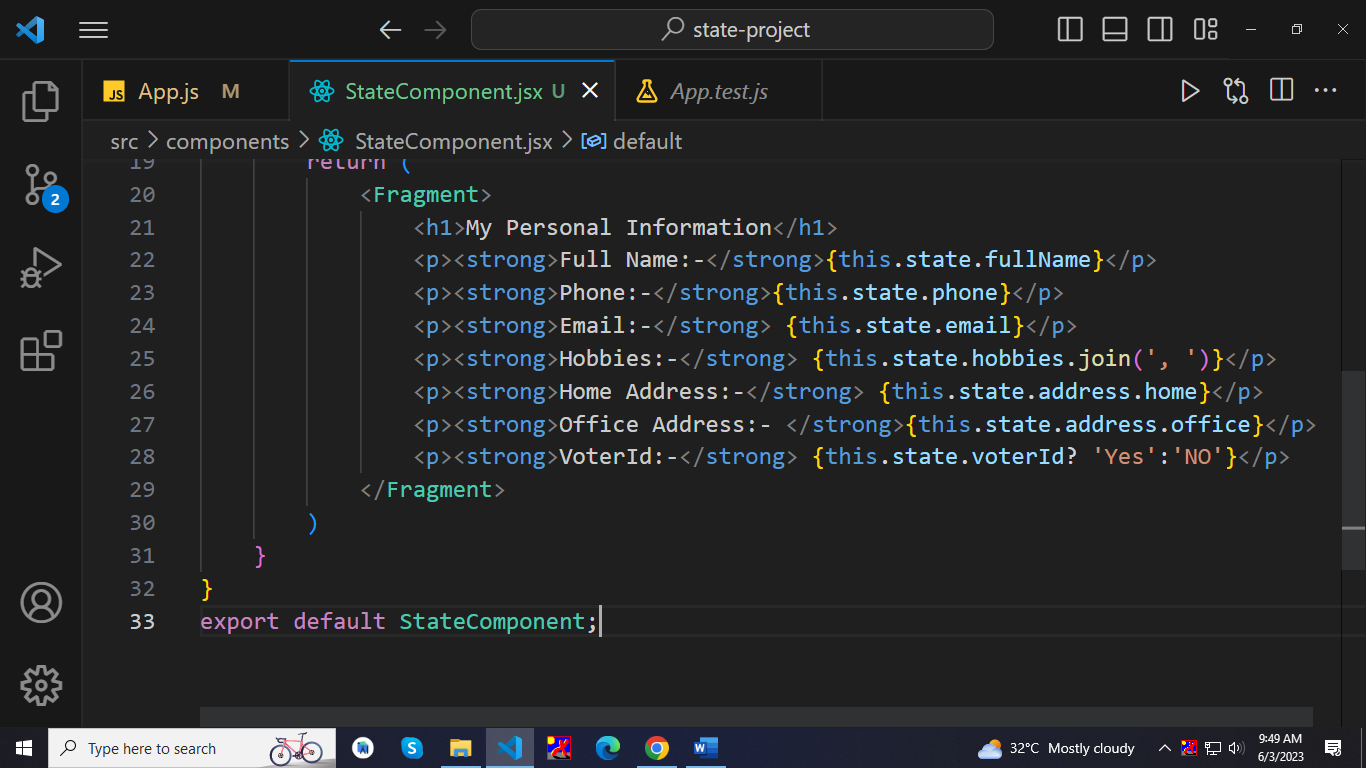
    }

}

export default StateComponent;







What is Props

Props is shorthand of properties

It is pass data from parent to child

It is immutable(means it can not change by child)

Props pass as HTML attribute or Element and receive as a object

**Props Component simple receive**

import React, {Component,Fragment} from 'react'

class PropsComponent extends Component{

    render(){

        console.log(this.props);

        return(

           <Fragment>

                <h1>Props Component Simple recieve</h1>

               <p>{this.props.message}</p>

               <p>{this.props.array}</p>

               <p>{this.props.personInfo.firstName}</p>

           </Fragment>

        )

    }

}

export default PropsComponent

**When call**

<PropsComponent message={messagePass} array={array} personInfo={personInfo}/>

**Props Component Destructuring**

import React, {Component,Fragment} from 'react'

class PropsDestructuring extends Component{

    render(){

        const {firstName,lastName} =this.props.personInfo;

        return(

           <Fragment>

            <h1>Props Component destructuring recieve</h1>

               <p>{firstName}</p>

               <p>{lastName}</p>

           </Fragment>

        )

    }

}

export default PropsDestructuring

<PropsDestructuring message={messagePass} array={array} personInfo={personInfo}/>

**Props Component Element**

import React, {Component,Fragment} from 'react'

class PropsElementPass extends Component{

    render(){

        return(

           <Fragment>

            <h1>Props Component Element recieve</h1>

               <p>{this.props.children}</p>

           </Fragment>

        )

    }

}

export default PropsElementPass

**App.js call**

<PropsElementPass>

        <p>Hello This Is paragraph</p>

      </PropsElementPass>

**Function Props Simple**

import React,{Fragment} from "react"

const FunctionProps = (props) => {

    return (

        <Fragment>

            <h1>Function Props Pass simple</h1>

            <p>{props.message}</p>

            <p>{props.array}</p>

            <p>{props.personInfo.firstName}</p>

        </Fragment>

    )

}

export default FunctionProps

<FunctionProps message={messagePass} array={array} personInfo={personInfo} />

**Function Props Destructuring**

import React,{Fragment} from "react"

const FunctionPropsDestructuring = ({message}) => {

    return (

        <Fragment>

            <h1>Function Props parameter destructuring</h1>

            <p>{message}</p>

        </Fragment>

    )

}

export default FunctionPropsDestructuring

<FunctionPropsDestructuring message={messagePass} array={array} personInfo={personInfo} />

**Function props seprate destructuring**

import React,{Fragment} from "react"

const FunctionPropsDestructuringTwo = (props) => {

    const {message,array,personInfo} = props;

    return (

        <Fragment>

            <h1>Function Props Destructuring Two</h1>

            <p>{message}</p>

            <p>{array}</p>

            <p>{personInfo.firstName}</p>

        </Fragment>

    )

}

export default FunctionPropsDestructuringTwo

<FunctionPropsDestructuringTwo message={messagePass} array={array} personInfo={personInfo} />

**Function props element pass**

import React,{Fragment} from "react"

const FunctionPropsElement = (props) => {

    return (

        <Fragment>

            <h1>Function Props Pass Element</h1>

            <div>{props.children}</div>

        </Fragment>

    )

}

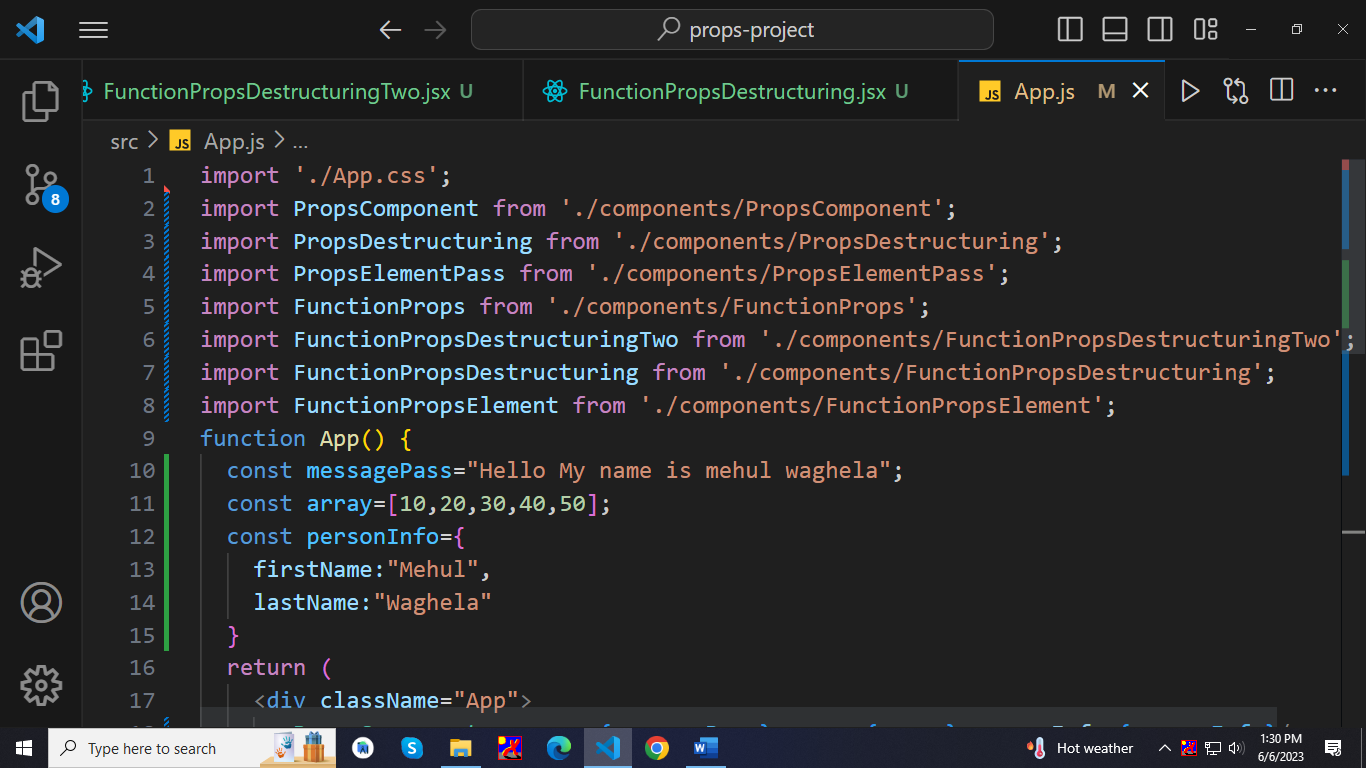
export default FunctionPropsElement

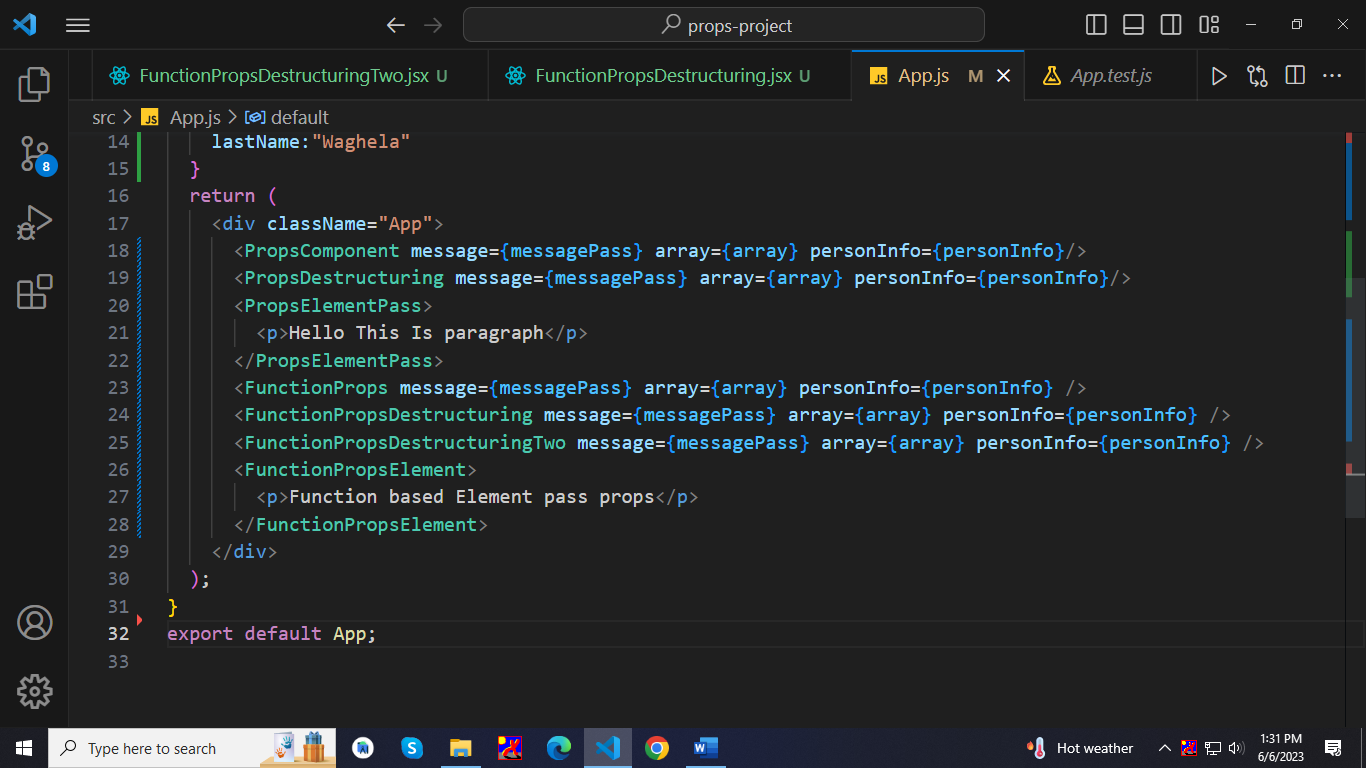
<FunctionPropsElement>

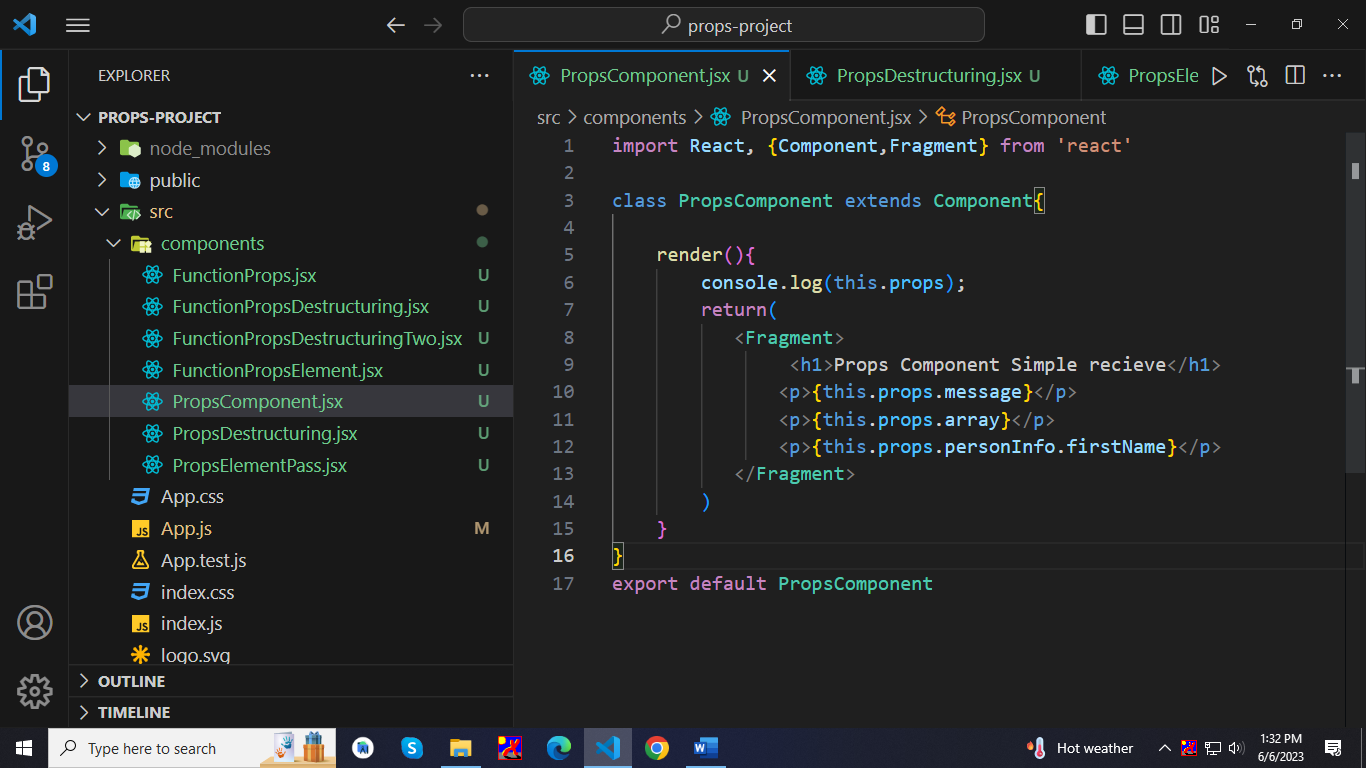
        <p>Function based Element pass props</p>

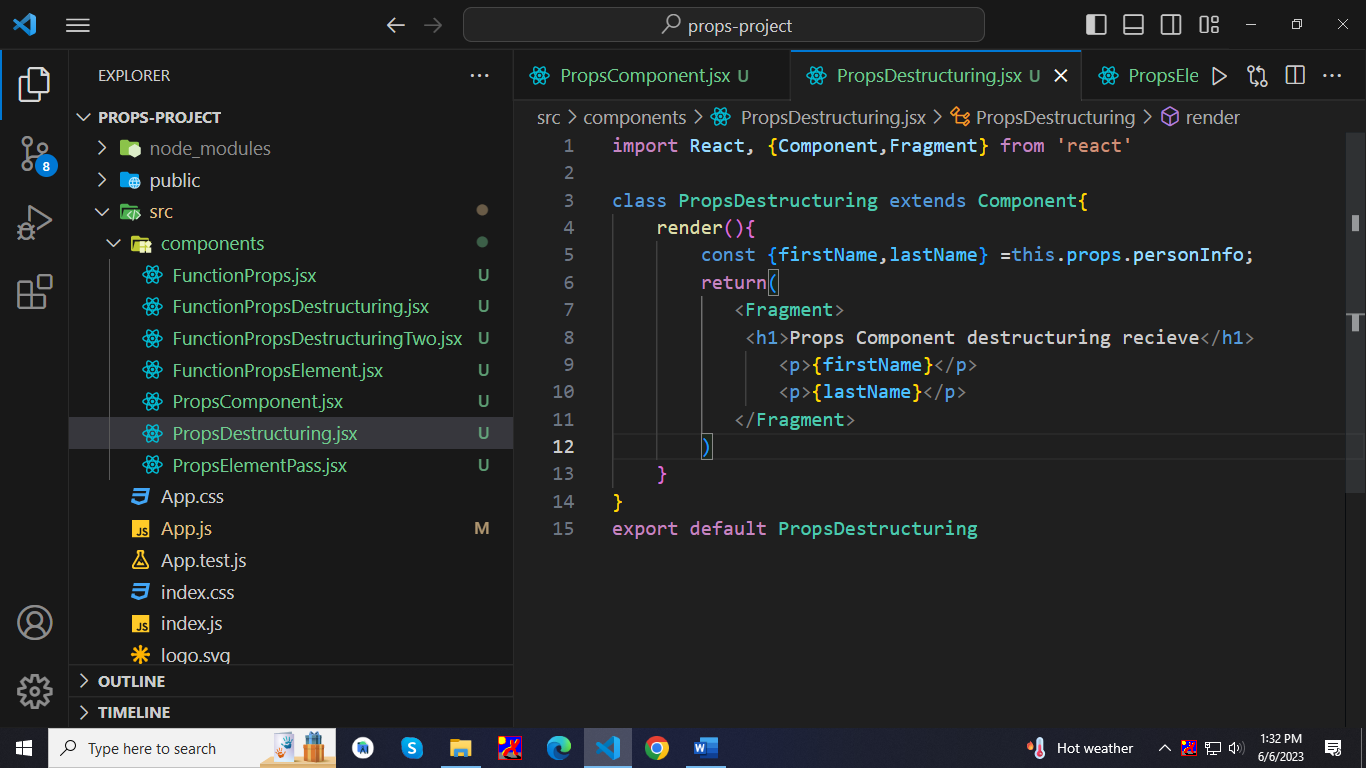
      </FunctionPropsElement>

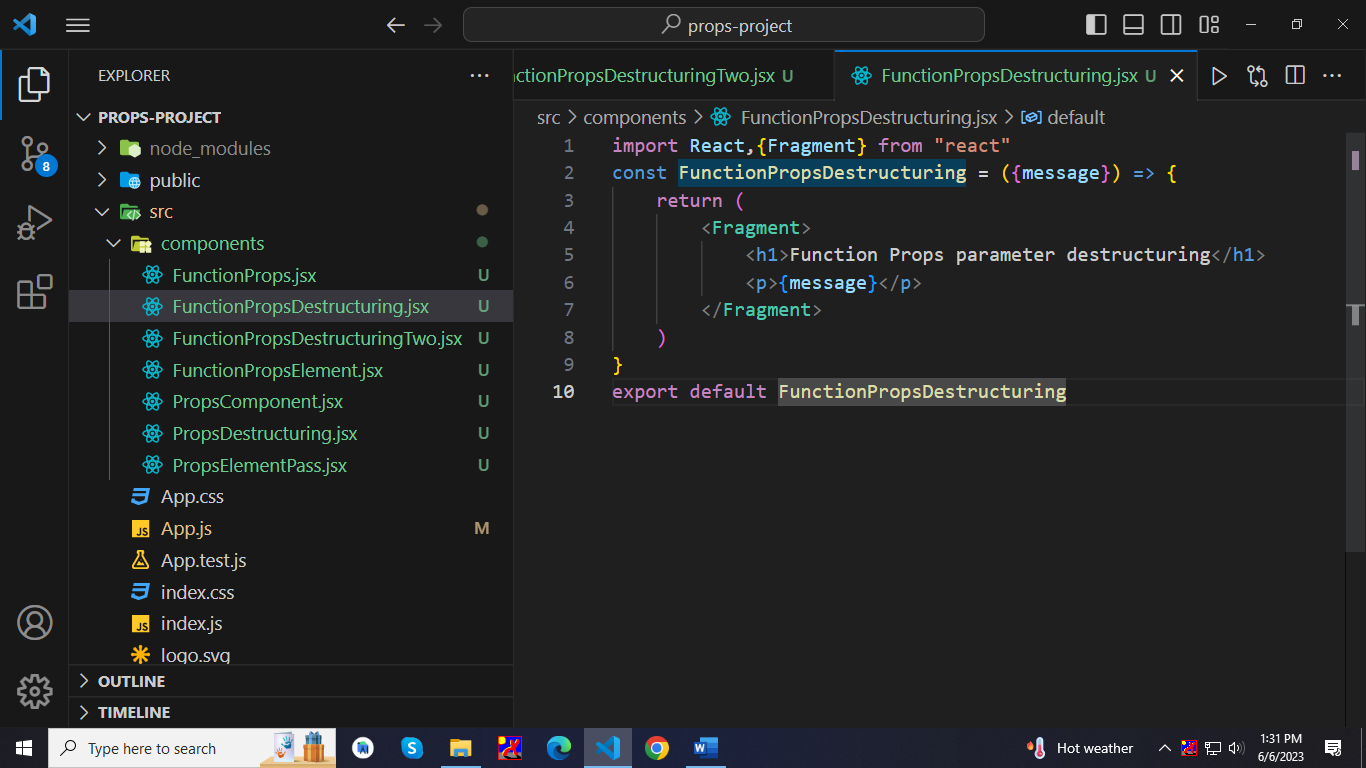
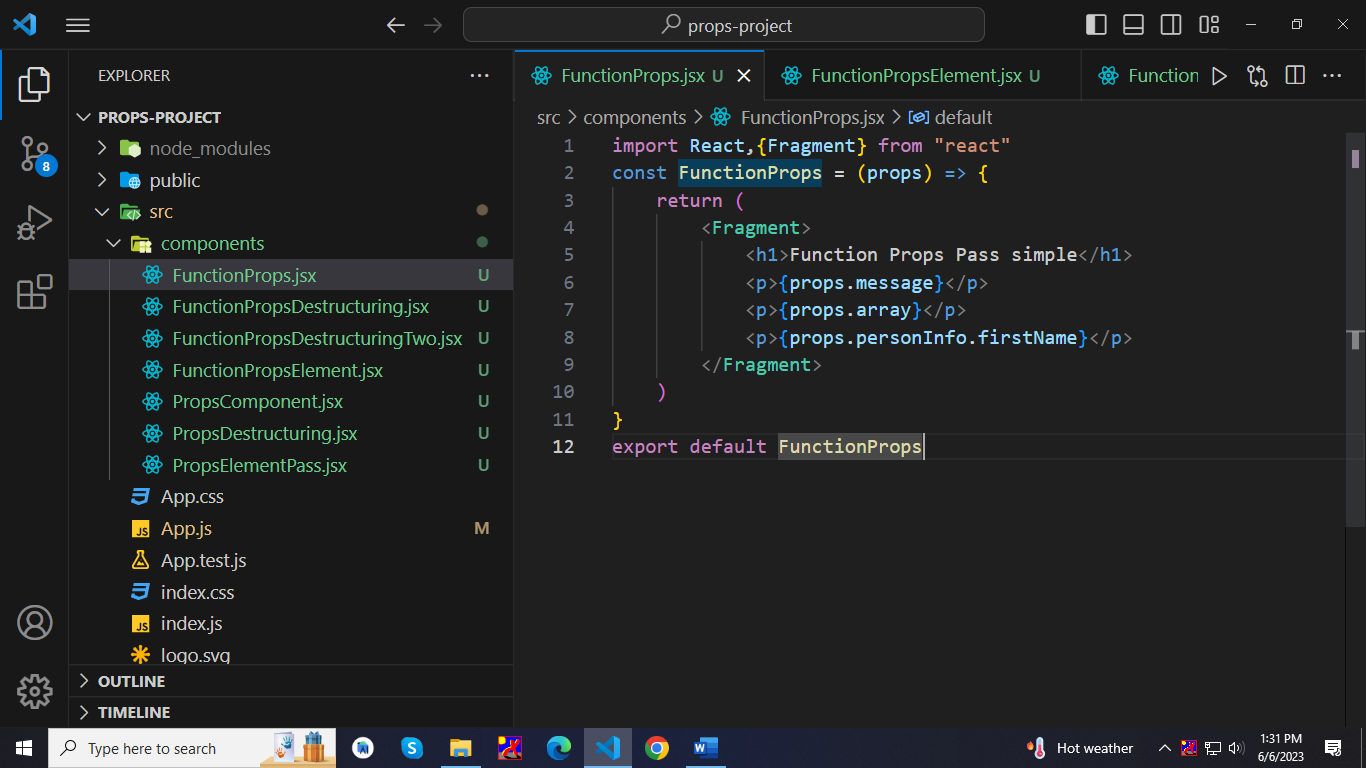
**App.js**

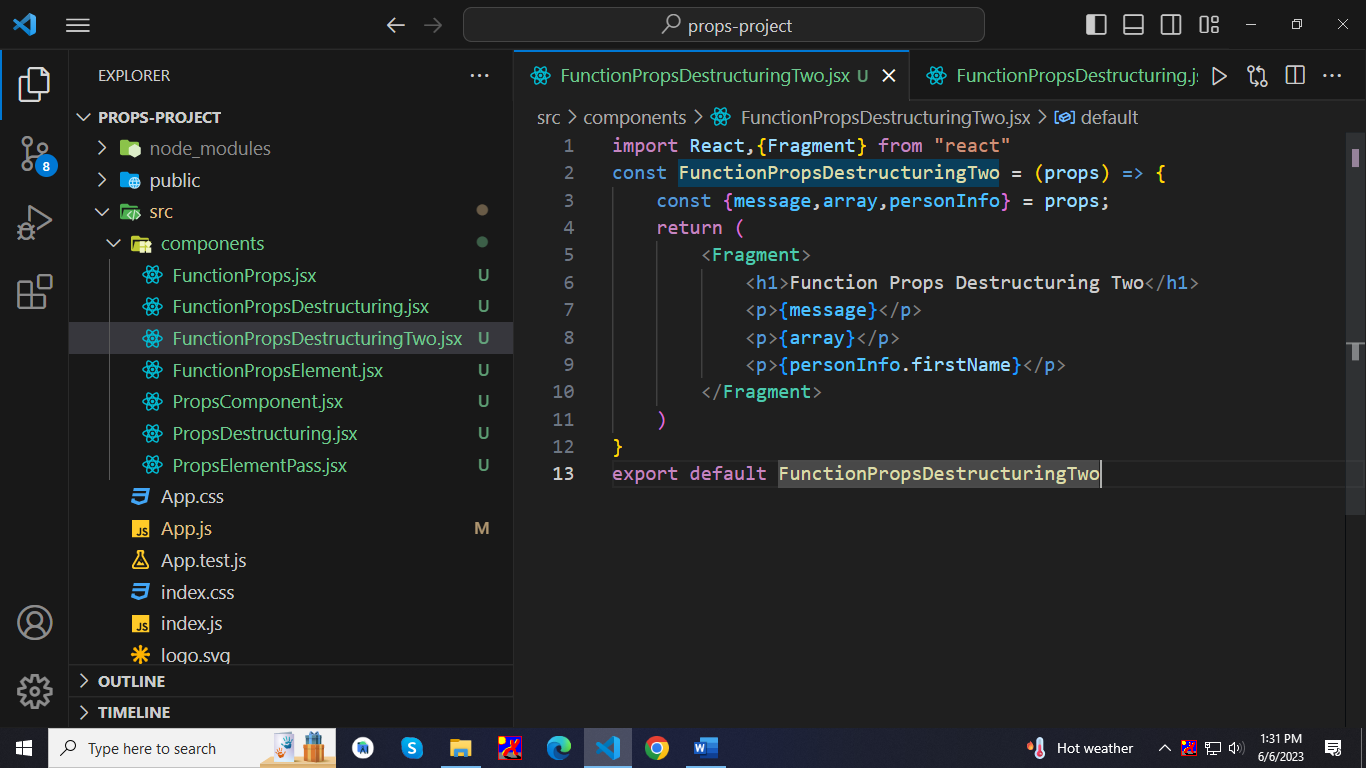


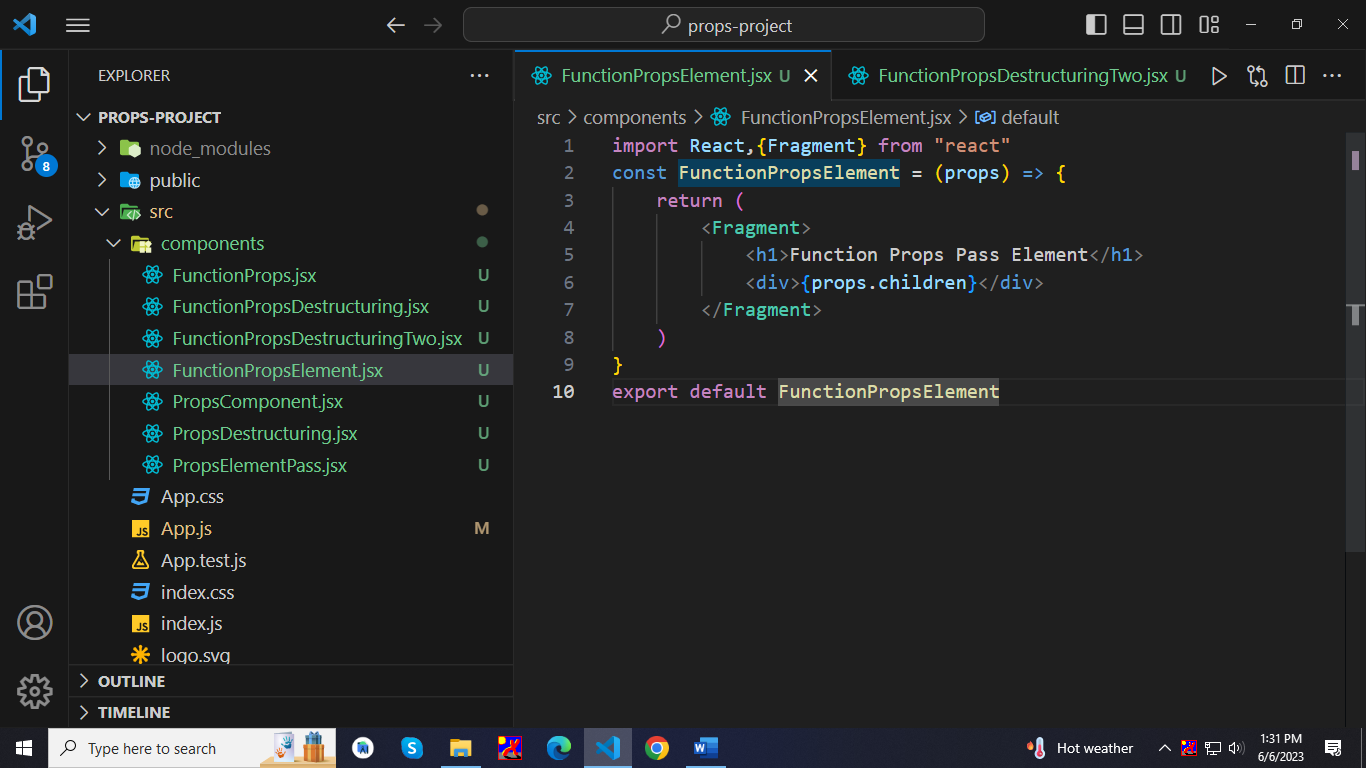












**Props Task**

**Send Data parent to child and set state from child**

**Parent component**

import React, { Component, Fragment } from 'react'

import ChildComponent from './ChildComponent'

class ParentComponent extends Component {

  constructor() {

    super()

    this.state = {

      welcome: "Guest User",

      message: "Please sign in",

      flag: false

    }

  }

  show = () => {

    this.setState({ flag: !this.state.flag })

    if (this.state.flag) {

      this.setState({

        welcome: "Welcome guest user",

        message: "Please login"

      })

    }

    else {

      this.setState({

        welcome: "Welcome Mehul waghela",

        message: "Congratulations"

      })

    }

  }

  render() {

    return (

      <Fragment>

        <ChildComponent welcome=

          {this.state.welcome}

          message={this.state.message}

          flag={this.state.flag}

          show={this.show} />

      </Fragment>

    )

  }

}

export default ParentComponent

**Child Component**

import React, { Component, Fragment } from 'react'

class ChildComponent extends Component {

  render() {

    const { welcome, message, flag,show} = this.props;

    console.log(this.props)

    return (

      <Fragment>

        <h1>Profile</h1>

        <p>{welcome}</p>

        <p>{message}</p>

        <p>{flag}</p>

        <button onClick={show}>{flag? "Offline" : "Online"}</button>

      </Fragment>

    )

  }

}

export default ChildComponent

**Listing-Task**

import {

    Card,

    CardBody,

    CardSubtitle,

    CardText,

    Button,

    CardTitle,

    Container,

    Row,

    Col

} from 'reactstrap'

import imgOne from '../assets/images/one.jpg'

import imgTwo from '../assets/images/two.jpg'

import imgThree from '../assets/images/three.jpg'

import imgFour from '../assets/images/four.jpg'

import imgFive from '../assets/images/five.jpg'

import imgSix from '../assets/images/six.jpg'

import imgSeven from '../assets/images/seven.jpg'

import imgEight from '../assets/images/eight.jpg'

import imgNine from '../assets/images/nine.jpg'

import React, { Component, Fragment } from 'react'

class CardComponent extends Component {

    constructor() {

        super()

        this.state = {

            personalInfo: [

                {

                    firstName: "Mehul",

                    lastName: "Waghela",

                    info: "The building block of a card is the CardBody Use it whenever you need a padded section within a",

                    img: imgOne

                },

                {

                    firstName: "Pratik",

                    lastName: "Chasiya",

                    info: "Card titles are used by adding CardTitle with an optional h\* prop. In the same way, links are added",

                    img: imgTwo

                },

                {

                    firstName: "Bhargav",

                    lastName: "Borad",

                    info: "Create lists of content in a card with a flush list group.Create lists of content in a card with a flush ",

                    img: imgThree

                },

                {

                    firstName: "Milan",

                    lastName: "Kathiriya",

                    info: "Lorem ipsum dolor, sit amet consectetur adipisicing elit. Molestiae, harum.",

                    img: imgFour

                },

                {

                    firstName: "Sagar",

                    lastName: "Davara",

                    info: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolore, molestias",

                    img: imgFive

                },

                {

                    firstName: "Darshan",

                    lastName: "Patel",

                    info: "Some quick text to build on the card title and make up the bulk of the card‘s content.",

                    img: imgSix

                },

                {

                    firstName: "Harsh",

                    lastName: "Bhalala",

                    info: "Headquartered in Mumbai, Sun Infosystems was founded by Mr. Sachin Vira, who ",

                    img: imgSeven

                },

                {

                    firstName: "Pratham",

                    lastName: "Patel",

                    info: " Mr. Sachin’s functional expertise spreads into handling several business functions such ",

                    img: imgEight

                },

                {

                    firstName: "Saimil",

                    lastName: "Patel",

                    info: " Mr. Sachin’s functional expertise spreads into handling several business functions such ",

                    img: imgNine

                }

            ]

        }

    }

    render() {

        return (

            <Fragment>

                <Container>

                    <Row className='g-1'>

                        {this.state.personalInfo.map((item) => {

                            return (

                                <Col sm={12} md={6} lg={4} className='g-0 p-1'>

                                    <Card

                                        color="secondary"

                                        outline

                                        style={{

                                            height:"450px"

                                        }}

                                    >

                                        <img

                                            alt="Sample"

                                            src={item.img}

                                            height="250px"

                                        />

                                        <CardBody className='g-0'>

                                            <CardTitle tag="h5">

                                                {item.firstName}

                                            </CardTitle>

                                            <CardSubtitle

                                                className="mb-2 text-muted"

                                                tag="h6"

                                            >

                                                {item.lastName}

                                            </CardSubtitle>

                                            <CardText>

                                                {item.info}

                                            </CardText>

                                            <Button>

                                                Button

                                            </Button>

                                        </CardBody>

                                    </Card>

                                </Col>

                            )

                        })}

                    </Row>

                </Container>

            </Fragment>

        )

    }

}

export default CardComponent

**Lifecycle component Mounting Phase**

**Child code**

import { Component } from "react";

class LifeCycleComponent extends Component {

    constructor() {

        super()

        this.state = {

            fullName: null,

            age: null,

            address: null,

        }

    }

    /\*getDerivedStateFromProps  this process complete before rendering ,render not call before this function execution done

    getDerivedStateFromProps is natural space for  setstate by props ,if we not set state by props and directly set state data then it is not natural space of update

    Aapre ahiye je pan process karvi hoi a kari ahakiye,pan je kaam render pehla karvu hoy a ahiya lakhvu\*/

    static getDerivedStateFromProps(props, state) {

        return {

            fullName: props.fullName,

            age: props.age,

            address: props.address

        };

    }

    // this is not natural space due to we set data directly not by props

    // static getDerivedStateFromProps(props, state) {

    //     return {

    //         fullName: "mehul",

    //     };

    // }

    render() {

        return (

            <div>

                <h1>Full Name={this.state.fullName}</h1>

                <h1>Age={this.state.age}</h1>

                <h1>Address={this.state.address}</h1>

            </div>

        )

    }

    /\*componentDidMount is run only after render function execution complete

    Aapre ahiye je pan process karvi hoi a kari ahakiye,pan je kaam render pachi karvu hoy a ahiya lakhvu

    setstate j karvu jaruri nathi\*/

    componentDidMount() {

        this.setState({ address: 'oscar Career Point' });

    }

}

export default LifeCycleComponent

**Parent code**

import { Component } from "react";

import LifeCycleComponent from "./LifeCycleComponent";

class ParentComponent extends Component {

    constructor() {

        super()

        this.state = {

            fullName: "Mehul Waghela",

            age:27,

            address:"71,Deepnagar Society,kim"

        }

    }

    render() {

        return (

            <LifeCycleComponent fullName={this.state.fullName} age={this.state.age} address={this.state.address} />

        )

    }

}

export default ParentComponent

**App.js code**

import './App.css';

import ParentComponent from './components/ParentComponenent';

function App() {

  return (

    <div className="App">

      <ParentComponent />

    </div>

  );

}

export default App;

**Life cycle Class Update Phase**

**Child Component**

import React, { Component, Fragment } from 'react'

class ChildComponent extends Component {

    constructor() {

        super()

        this.state = {

            firstName: null,

            age: null,

            login: null,

            newage: null,

            prevAge: null,

        }

    }

    /\* 1=>Mounting Phase \*/

    /\* this function always run before render \*/

    /\* After render COmponentDidMount function Call if we do any changes or update any state or anything else then updating phase start\*/

    /\* 2=>Updating Phase \*/

    /\* 5 method of updating phase

       1-getDerivedStateFromProps()

       2-shouldComponentUpdate()

       3-render()

       4-getSnapshotBeforeUpdate()

       5-componentDidUpdate() \*/

    /\* 1-getDerivedStateFromProps()=> if we update anything after render then updating phase start

    in updating phase first getDerivedStateFromProps (this is function not new function

    this is same function which we make in mounting phase)

    function call and then rerendering start and ComponentDidUpdate function call

    \*/

    /\*

    2-shouldComponentUpdate()=> this method is use to stop rendering

        if we not write this function then it is return true by default

        if we write this function then we have to return true or false

        if we return false then rendering is stop

        we can stop render on perticular condition using this function

     \*/

    /\* 4-getSnapshotBeforeUpdate() => This method use when we want prev state or prev props which we change in any method.  \*/

    /\* 5-componentDidUpdate() => THis function call after rerendering \*/

    static getDerivedStateFromProps(props, state) {

        return (

            ({

                firstName: props.firstName,

            })

        )

    }

    stopFun = () => {

        this.setState({ login: "false" })

    }

    render() {

        return (

            <Fragment>

                <h1>Personal Information</h1>

                <p>First name={this.state.firstName}</p>

                <p>Age={this.state.age}</p>

                <p>Login={this.state.login}</p>

                <p id="div1"></p>

                <button onClick={this.stopFun}>StopProcess</button>

            </Fragment>

        )

    }

    componentDidMount() {

        console.log("Component render");

        setTimeout(() => {

            this.setState({ age: 70 })

        }, 3000)

    }

    componentDidUpdate() {

        setTimeout(() => {

            this.setState({ login: "true", newage: 80 })

            console.log("Component rerender");

        }, 3000)

    }

    shouldComponentUpdate() {

        if (this.state.login === false) {

            return false

        }

        else {

            return true

        }

    }

    getSnapshotBeforeUpdate(prevProps, prevState) {

        if (prevState.age) {

            return (document.getElementById("div1").innerHTML =

                "Before the update, the age " + prevState.age)

        }

        else {

            return (document.getElementById("div1").innerHTML =

                "Before the update, the age " + prevState.age)

        }

    }

}

export default ChildComponent

**Parent Component**

import React, {Component, Fragment} from 'react'

import ChildComponent from './ChildComponent'

class ParentComponent extends Component{

    constructor(){

        super()

        this.state={

            firstName:"Mehul",

            age:27

        }

    }

    render()

    {

        return(

            <Fragment>

                <ChildComponent firstName={this.state.firstName} age={this.state.age} />

            </Fragment>

        )

    }

}

export default ParentComponent

**Form handling**

import { Component, Fragment } from "react";

import { Form, Input, Container, Label, Row, Col } from "reactstrap";

class FormHandling extends Component {

    constructor() {

        super()

        this.state = {

            profile: {

                userName: '',

                age: null

            }

        }

    }

    /\* 1=>Native Event is default Event browser

    like button =>button event click is native event \*/

    /\* 2=>Synthetic Event is occur when we call function using any event

    like onChange or onInput etc.. and that event occur in parameter of that function \*/

    onInputChange = (e) => {

        this.setState({

            /\* in crud we have multiple input and if we want all value access by one function then we can access define below \*/

            /\*...this.state.profile =>this is spread operator which we use to set profile same as above,and e.target.value set that key only which input we change  \*/

            /\* if we not use spread operator then which we input change that key set and other item remove from that object.

             in this  profile: {

                ...this.state.profile, [e.target.name]: e.target.value

            }  we set profile key

            if we write profile:{

                                     }

            like this, it is blank profile. If we not write spread operator then profile become blank and we have to set previous data that place and we have to only change that value which input we change \*/

            profile: {

                ...this.state.profile, [e.target.name]: e.target.value

            }

        }, () => {

            console.log(e);

            console.log(this.state.profile);

        })

    };

    render() {

        return (

            <Fragment>

                <Container>

                    <Form>

                        <Row>

                            <Col>

                                <Label for="userName">Username</Label>

                                {/\* onchange event call function when we change in input \*/}

                                <Input onChange={this.onInputChange} name="userName" id="userName" bsSize="" />

                            </Col>

                            <Col>

                                <Label for="age">Age</Label>

                                <Input onChange={this.onInputChange} type="number" name="age" id="age" bsSize="" />

                            </Col>

                        </Row>

                    </Form>

                </Container>

            </Fragment>

        )

    }

}

export default FormHandling

**Pure Component in Class**

import React, { PureComponent } from 'react'

/\* components that do not re-render when the value of props and state has been updated with the same values. \*/

/\* PureComponent is use for stop unnecessary rendering

if we set state and that state is same as previous state then output after rendering same.

But it is unnecessary rendering.We know that after rerendering same output occur and it take time to render by stop this process we can improve performance

To make PureComponent  like below

after extends use PureComponent instead of Component

Pure Component is use only for Class Component\*/

export class MainComponent extends PureComponent {

    render() {

        console.log("Pure Component");

        return (

            <div>{this.props.message}</div>

        )

    }

}

export default MainComponent

**Parent component**

import React, { Component, Fragment } from 'react'

import ChildComponent from './ChildComponent'

import MainComponent from './MainComponent'

export class ParentComponent extends Component {

    constructor() {

        super()

        this.state = {

            message: "Hello World"

        }

    }

    /\* in this code we set state same Hello world therefore firsttime render and then render stop because set state message is same as original state \*/

    componentDidMount() {

        setInterval(() => {

            this.setState({ message: "Hello World" });

        }, 3000);

    }

    render() {

        console.log("Parent Component");

        return (

            <Fragment>

                <ChildComponent message={this.state.message} />

                <MainComponent message={this.state.message} />

            </Fragment>

        )

    }

}

export default ParentComponent

**Child Component**

import React, { Component } from 'react'

export class ChildComponent extends Component {

    render() {

        console.log("Child Component");

        return (

            <div>{this.props.message}</div>

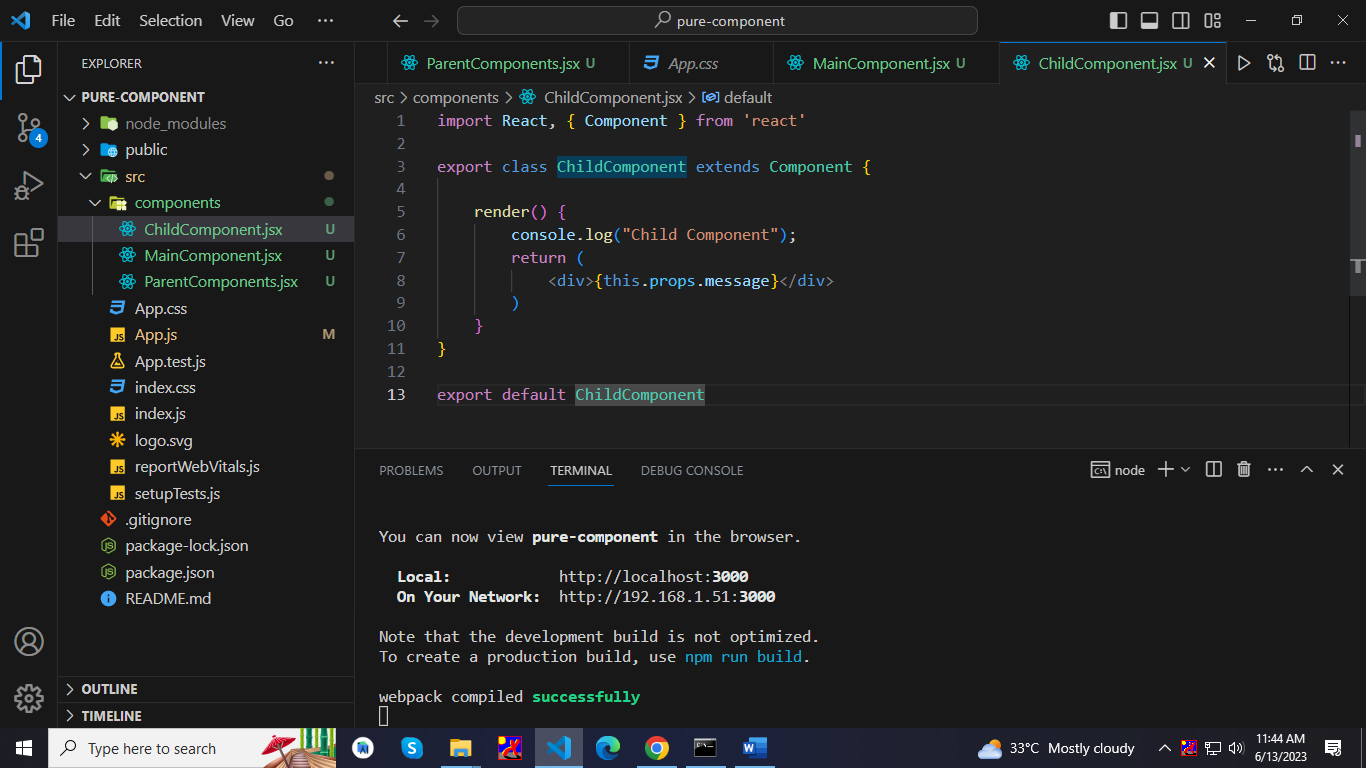
        )

    }

}

export default ChildComponent

**Folder structure for Pure Component**



**Memo Function Component**

**Parent Component**

import React, { Component, Fragment } from 'react'

import ChildComponent from './ChildComponent'

import MainComponent from './MainComponent'

class ParentComponent extends Component {

    constructor() {

        super()

        this.state = {

            message: "Hello World"

        }

    }

    /\* in this code we set state same Hello world therefore firsttime render and then render stop because set state message is same as original state \*/

    componentDidMount() {

        setInterval(() => {

            this.setState({ message: "Mehul Waghela" });

        }, 3000);

    }

    render() {

        console.log("Parent Component");

        return (

            <Fragment>

                <ChildComponent message={this.state.message} />

                <MainComponent message={this.state.message} />

            </Fragment>

        )

    }

}

export default ParentComponent

**Memo Component**

import { memo } from 'react'

/\* Compulsory use in curly bracket {memo} \*/

/\* components that do not re-render when the value of props and state has been updated with the same values. \*/

/\* PureComponent is use for stop unnecessary rendering

if we set state and that state is same as previous state then output after rendering same.

But it is unnecessary rendering.We know that after rerendering same output occur and it take time to render by stop this process we can improve performance

To make PureComponent  like below

export default memo(MainComponent) instead of export default MainComponent

Memo is use only for Function Component\*/

const MainComponent = ({ message }) => {

    console.log("Memo Component");

    return (

        <h1>Memo Component={message}</h1>

    )

}

export default memo(MainComponent)

**Child Component**

const ChildComponent = ({ message }) => {

    console.log("Child Component");

    return (

        <h1>Child Component={ message }</h1>

    )

}

export default ChildComponent

**Conditional Rendering**

**If Else**

import { Component, Fragment } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            age: 18

        }

    }

    /\* We can show and hide html on particular condition using if else \*/

    render() {

        if (this.state.age > 18) {

            return (

                <Fragment>

                    <h1>You Can Vote</h1>

                </Fragment>

            )

        }

        else {

            return (

                <Fragment>

                    <h1>You Can Not Vote</h1>

                </Fragment>

            )

        }

    }

    componentDidMount() {

        setTimeout(() => {

            this.setState({ age: 20 })

        }, 3000);

    }

}

export default Conditional

**Ternary**

import { Component } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            age: 18

        }

    }

    /\* We can show and hide html on particular condition using Condition \*/

    render() {

        return (this.state.age > 18) ? <h1> You Can Vote</h1> : <h1>You Can not Vote</h1>

    }

    componentDidMount() {

        setTimeout(() => {

            this.setState({ age: 20 })

        }, 3000);

    }

}

export default Conditional

**Logical**

import { Component, Fragment } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            age: 20

        }

    }

    /\* We can show and hide html on particular condition using logical \*/

    render() {

        return (

            <Fragment>

                <h1>Hello My name is mehul waghela</h1>

                {

                    this.state.age > 18 && <p>you can vote</p>

                }

                {

                    this.state.age <= 18 && <p>you can not vote</p>

                }

            </Fragment>

        )

    }

    componentDidMount() {

        setTimeout(() => {

            this.setState({ age: 18 })

        }, 3000);

    }

}

export default Conditional

**Conditional Rendering Task**

import { Component, Fragment } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            online: false,

        }

    }

    changestate = () => {

        this.setState({ online: !this.state.online })

    }

    /\* We can show and hide html on particular condition using logical \*/

    render() {

        return (

            <Fragment>

                <h1>Hello My name is mehul waghela</h1>

                {

                    this.state.online === true &&

                    <div>

                        <h2>Welcome mehul waghela </h2>

                        <p>Congratulation You are logged in</p>

                    </div>

                }

                {

                    this.state.online === false &&

                    <div>

                        <h2>Guest user</h2>

                        <p>Please login</p>

                    </div>

                }

                <button onClick={this.changestate}>{this.state.online ? "Offline" : "Online"}</button>

            </Fragment>

        )

    }

    componentDidMount() {

        setTimeout(() => {

            this.setState({ age: 18 })

        }, 3000);

    }

}

export default Conditional

**If Else**

import { Component, Fragment } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            online: false,

        }

    }

    changestate = () => {

        this.setState({ online: !this.state.online })

    }

    /\* We can show and hide html on particular condition using logical \*/

    render() {

        return (

            <Fragment>

                <h1>Hello My name is mehul waghela</h1>

                {

                    this.state.online === true &&

                    <div>

                        <h2>Welcome mehul waghela </h2>

                        <p>Congratulation You are logged in</p>

                    </div>

                }

                {

                    this.state.online === false &&

                    <div>

                        <h2>Guest user</h2>

                        <p>Please login</p>

                    </div>

                }

                <button onClick={this.changestate}>{this.state.online ? "Offline" : "Online"}</button>

            </Fragment>

        )

    }

    componentDidMount() {

        setTimeout(() => {

            this.setState({ age: 18 })

        }, 3000);

    }

}

export default Conditional

**Ternary**

import { Component, Fragment } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            online: false,

        }

    }

    changestate = () => {

        this.setState({ online: !this.state.online })

    }

    /\* We can show and hide html on particular condition using Ternary \*/

    render() {

        return this.state.online ?

            <div>

                <h2>Welcome mehul waghela </h2>

                <p>Congratulation You are logged in</p>

            </div>

            : <div>

                <h2>Guest user</h2>

                <p>Please login</p>

            </div>

    }

}

export default Conditional

**Conditional Rendering Task**

import { Component, Fragment } from 'react'

class Conditional extends Component {

    constructor() {

        super()

        this.state = {

            online: false,

        }

    }

    changestate = () => {

        this.setState({ online: !this.state.online })

    }

    /\* We can show and hide html on particular condition using logical \*/

    render() {

        return (

            <Fragment>

                <h1>Hello My name is mehul waghela</h1>

                {

                    this.state.online === true &&

                    <div>

                        <h2>Welcome mehul waghela </h2>

                        <p>Congratulation You are logged in</p>

                    </div>

                }

                {

                    this.state.online === false &&

                    <div>

                        <h2>Guest user</h2>

                        <p>Please login</p>

                    </div>

                }

                <button onClick={this.changestate}>{this.state.online ? "Offline" : "Online"}</button>

            </Fragment>

        )

    }

    componentDidMount() {

        setTimeout(() => {

            this.setState({ age: 18 })

        }, 3000);

    }

}

export default Conditional

**Bind Method**

import { Component, Fragment } from "react";

class Myfunction extends Component {

    constructor() {

        super()

        this.state = {

            fullname: 'Mehul'

        }

        /\* in this program we bind this in sum function. If we not do this, then we can not use 'this' in sum function \*/

        /\* if we use arrow function then this problem by default solve in this es6 arrow method \*/

        this.sum = this.sum.bind(this)

    }

    sum() {

        this.setState({ fullname: 'pratik' })

    }

    render() {

        return (

            <Fragment>

                <h1>{this.state.fullname}</h1>

                <button onClick={this.sum}>Good Morning</button>

            </Fragment >

        )

    }

}

export default Myfunction

**Class Component Crud**

import '../styles/header.css'

import { Button, Modal, ModalHeader, ModalBody, ModalFooter, Container, Label, Input, Form, Row, Col, Table } from 'reactstrap';

import { Component, Fragment } from 'react';

class Header extends Component {

    constructor(props) {

        super(props);

        this.state = {

            modal: false,

            backdrop: false,

            profile: {

                id: '',

                firstName: '',

                lastName: '',

                userName: '',

                email: '',

                password: '',

                age: null,

                phoneNo: null,

                gender: '',

                hobby: []

            },

            allstudent: [],

            cPassword: '',

            editId: null

        }

    }

    /\* Add function \*/

    addFunction = () => {

        this.formReset()

        this.toggle()

        this.setState({ editId: null })

    }

    /\* to open Modal \*/

    toggle = () => {

        this.setState({

            modal: !this.state.modal,

        });

        if ((!this.state.modal) && this.state.editId === null) {

            this.formReset()

        }

    }

    /\* input field change function \*/

    onInputChange = (e) => {

        this.setState({

            profile: {

                ...this.state.profile, [e.target.name]: e.target.value, id: (this.state.editId !== null) ? this.state.editId : Date.now()

            }

        })

    }

    /\* hobby field change function \*/

    onHobbyChange = (e) => {

        if (e.target.checked) {

            this.setState({

                profile: {

                    ...this.state.profile, hobby: [...this.state.profile.hobby, e.target.checked && e.target.value]

                }

            })

        }

        else {

            this.setState({

                profile: { ...this.state.profile, hobby: this.state.profile.hobby.filter((item) => item !== e.target.value) }

            })

        }

    }

    /\* On confirm password \*/

    confirmPassword = (e) => {

        this.setState({

            cPassword: e.target.value

        })

    }

    /\* get local storage function \*/

    getLocal = () => {

        this.setState({ allstudent: JSON.parse(localStorage.getItem("allstudent")) }, () => {

        })

    }

    /\* set local storage function \*/

    setLocal = () => {

        localStorage.setItem("allstudent", JSON.stringify(this.state.allstudent));

    }

    /\* submit function \*/

    insertRow = (e) => {

        e.preventDefault();

        if (this.state.editId === null && (this.state.profile.password !== this.state.cPassword)) {

            alert("Password and Confirm Password Must be Same")

        }

        else {

            if (this.state.editId !== null) {

                this.state.allstudent.splice(this.state.allstudent.findIndex((item) => item.id === this.state.editId), 1, this.state.profile)

                this.setState({

                    editId: null

                })

                this.setLocal();

                this.getLocal();

                this.formReset();

            }

            else {

                this.setState({

                    allstudent:

                        [...this.state.allstudent, this.state.profile]

                }, () => {

                    this.setLocal();

                    this.getLocal();

                    this.formReset();

                })

            }

        }

    }

    componentDidMount() {

        if (JSON.parse(localStorage.getItem("allstudent")) !== null) {

            this.getLocal()

        }

    }

    /\* Edit Function \*/

    editRow = (id) => {

        this.setState({ editId: id, profile: this.state.allstudent.filter((item) => item.id === id)[0] }, () => {

            this.toggle()

        })

    }

    /\* Delete Function \*/

    deleteRow = (id) => {

        if (window.confirm("Are You Sure You Want to Delete This Row")) {

            this.state.allstudent.splice(this.state.allstudent.findIndex((elment) => elment.id === id), 1)

            this.setLocal();

            this.getLocal();

            this.formReset();

        }

    }

    /\* Form reset function \*/

    formReset = () => {

        this.setState({

            profile: {

                id: '',

                firstName: '',

                lastName: '',

                userName: '',

                password: '',

                email: '',

                age: null,

                phoneNo: null,

                gender: '',

                hobby: []

            },

        })

    }

    render() {

        return (

            <Fragment>

                <Container fluid className='bg-dark'>

                    <Container>

                        <div className='d-flex justify-content-between py-3'>

                            <h1 className='text-white'>STUDENT</h1>

                            <Button className='text-white rounded-pill px-5 fs-4' onClick={this.addFunction}>+ADD</Button>

                        </div>

                    </Container>

                </Container>

                <Modal isOpen={this.state.modal} toggle={this.toggle} className={this.props.className} backdrop={this.state.backdrop}>

                    <ModalHeader toggle={this.toggle}><p className='fw-bold fs-3'>Student Information</p></ModalHeader>

                    <Form id="form" onSubmit={this.insertRow}>

                        <ModalBody>

                            <Row className=''>

                                <Col xs={12} md={4}>

                                    <div className="mb-3">

                                        <Label for="InputOne" className='fw-bold'>First Name</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="text" id="InputOne"

                                            placeholder="First name"

                                            value={(this.state.profile.firstName) ?? ''}

                                            name="firstName" />

                                    </div>

                                    <div className="mb-3">

                                        <Label for="InputFive" className='fw-bold'>Phone No</Label>

                                        <Input

                                            type="tel" id="InputFive"

                                            placeholder="Your Mobile No"

                                            onChange={this.onInputChange}

                                            value={(this.state.profile.phoneNo) ?? ''}

                                            name="phoneNo" />

                                    </div>

                                    {

                                        (this.state.editId === null)

                                        &&

                                        <div className="mb-3">

                                            <Label for="InputSix" className='fw-bold'>Password</Label>

                                            <Input

                                                onChange={this.onInputChange}

                                                type="password"

                                                id="InputSix"

                                                placeholder="Password"

                                                name="password" />

                                        </div>

                                    }

                                    <div className="mt-2">

                                        <Label for="radioDiv" className='fw-bold'>Gender</Label>

                                        <div id="radioDiv" className="ps-2">

                                            <div className="form-check-inline">

                                                <Input

                                                    type="radio"

                                                    value="male"

                                                    id="radioOne"

                                                    name="gender"

                                                    onChange={this.onInputChange}

                                                    checked={this.state.profile.gender === 'male'}

                                                />

                                                <Label for="radioOne" className='ms-2'>Male</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    type="radio"

                                                    value="female"

                                                    id="radioTwo"

                                                    name="gender"

                                                    onChange={this.onInputChange}

                                                    checked={this.state.profile.gender === 'female'} />

                                                <Label for="radioTwo" className='ms-2'>Female</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    type="radio"

                                                    value="trans"

                                                    id="radioThree"

                                                    name="gender"

                                                    onChange={this.onInputChange}

                                                    checked={this.state.profile.gender === 'trans'} />

                                                <Label for="radioThree" className='ms-2'>Trans</Label>

                                            </div>

                                        </div>

                                    </div>

                                </Col>

                                <Col xs={12} md={4}>

                                    <div className="mb-3">

                                        <Label for="InputTwo" className='fw-bold'>Last Name</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="text"

                                            id="InputTwo"

                                            placeholder="Last name"

                                            value={(this.state.profile.lastName) ?? ''}

                                            name="lastName" />

                                    </div>

                                    <div className="mb-3">

                                        <Label for="InputThree" className='fw-bold'>Email address</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="email"

                                            id="InputThree"

                                            placeholder="Email"

                                            value={(this.state.profile.email) ?? ''}

                                            name="email" />

                                    </div>

                                    {

                                        (this.state.editId === null)

                                        &&

                                        <div className="mb-3">

                                            <Label for="InputSeven" className='fw-bold'>Confirm Password</Label>

                                            <Input

                                                onChange={this.confirmPassword}

                                                type="password" id="InputSeven"

                                                placeholder="Confirm Password"

                                                name="Cpass" />

                                        </div>

                                    }

                                    <div>

                                        <Label for="hobbyDiv" className='fw-bold'>Hobby</Label>

                                        <div id="hobbyDiv" className="ps-2">

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="cricket"

                                                    id="checkOne"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('cricket')}

                                                />

                                                <Label for="flexCheckDefault" className='ms-2'>Cricket</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="football"

                                                    id="checkTwo"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('football')} />

                                                <Label for="flexCheckDefault" className='ms-2'>Football</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="volleyball"

                                                    id="checkThree"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('volleyball')} />

                                                <Label for="checkThree" className='ms-2'>Volleyball</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="carom"

                                                    id="checkFour"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('carom')} />

                                                <Label for="checkFour" className='ms-2'>Carom</Label>

                                            </div>

                                        </div>

                                    </div>

                                </Col>

                                <Col xs={12} md={4}>

                                    <div className="mb-3">

                                        <Label for="InputNine" className='fw-bold'>User Name</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="text" id="InputNine"

                                            placeholder="User Name"

                                            value={(this.state.profile.userName) ?? ''}

                                            name="userName" />

                                    </div>

                                    <div className="mb-3">

                                        <Label for="InputFour" className='fw-bold'>Age</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="number"

                                            id="InputFour"

                                            placeholder="Your Age"

                                            value={(this.state.profile.age) ?? ''}

                                            name="age" />

                                    </div>

                                </Col>

                            </Row>

                        </ModalBody>

                        <ModalFooter>

                            <Button

                                type="submit"

                                className="btn-success"

                                id="submitBtn"

                                onClick={this.toggle}>Submit</Button>

                            <Button type="button" className="btn-danger" id="cancelBtn"

                                onClick={this.toggle}>Cancel</Button>

                        </ModalFooter>

                    </Form>

                </Modal>

                <Container fluid>

                    <Table responsive striped className='border border-1 '>

                        <caption className="fw-bolder text-center caption-top text-uppercase fs-3">Student personal information

                        </caption>

                        <thead id="thead" className="bg-dark">

                            <tr>

                                <th>SR. NO.</th>

                                <th>User name</th>

                                <th>First name</th>

                                <th>Last name</th>

                                <th>Email</th>

                                <th>Age</th>

                                <th>Telephone</th>

                                <th>Gender</th>

                                <th>Hobby</th>

                                <th>Action</th>

                            </tr>

                        </thead>

                        <tbody id="tbody">

                            {

                                (this.state.allstudent !== [])

                                &&

                                this.state.allstudent.map((item, index) => {

                                    return (

                                        <tr className="tr" key={index}>

                                            <th>{index + 1}</th>

                                            <th>{item.userName}</th>

                                            <th>{item.firstName}</th>

                                            <th>{item.lastName}</th>

                                            <th>{item.email}</th>

                                            <th>{item.age}</th>

                                            <th>{item.phoneNo}</th>

                                            <th>{item.gender}</th>

                                            <th>{item.hobby.join(',')}</th>

                                            <th>

                                                <Button className='me-2' onClick={() => this.editRow(item.id)}>Edit</Button>

                                                <Button onClick={() => this.deleteRow(item.id)}>Delete</Button>

                                            </th>

                                        </tr>

                                    )

                                })

                            }

                        </tbody>

                    </Table>

                </Container>

            </Fragment >

        )

    }

}

export default Header;

**CSS**

.fade {

    background-color: rgba(0, 0, 0, 0.8);

}

.modal {

    position: fixed;

    z-index: 999;

    top: 0;

    right: 0;

    left: 0;

    bottom: 0;

}

@media (max-width: 1500px) {

    .modal-dialog {

        max-width: 900px !important;

        margin-right: auto;

        margin-left: auto;

    }

}

**Unmounting of Class Component**

**Parent Component**

import { Component, Fragment } from 'react'

import ChildComponent from './ChildComponent'

class ParentComponent extends Component {

    constructor() {

        super()

        this.state = {

            delete: true

        }

    }

    deleteheader = () => {

        this.setState({ delete: !this.state.delete })

    }

    render() {

        let header;

        if (this.state.delete) {

            header = <ChildComponent />

        }

        return (

            <Fragment>

                {header}

                <h1>Parent Component</h1>

                <button onClick={this.deleteheader}>Delete Header</button>

            </Fragment>

        )

    }

}

export default ParentComponent

**Child Component**

import { Component, Fragment } from 'react'

class ChildComponent extends Component {

    /\* 3rd Phase=> Unmounting

    1 st Method=>componentwillmount

    This method is called when any part of component delete \*/

    componentWillUnmount() {

        alert("Header is deleted");

    }

    render() {

        return (

            <Fragment>

                <h1>Header</h1>

            </Fragment>

        )

    }

}

export default ChildComponent

**Class CRUD**

**Main component**

import logo from '../assets/logo.jpg'

import '../styles/header.css'

import { Button, Modal, ModalHeader, ModalBody, ModalFooter, Container, Label, Input, Form, Row, Col, Table } from 'reactstrap';

import { Component, Fragment } from 'react';

import Swal from 'sweetalert2';

class Header extends Component {

    constructor(props) {

        super(props);

        this.state = {

            modal: false,

            backdrop: false,

            profile: {

                id: '',

                firstName: '',

                lastName: '',

                userName: '',

                email: '',

                password: '',

                age: null,

                phoneNo: null,

                gender: '',

                hobby: []

            },

            allstudent: [],

            cPassword: '',

            editId: null,

        }

    }

    /\* Add function \*/

    addFunction = () => {

        this.formReset()

        this.toggle()

        this.setState({ editId: null })

    }

    /\* to open Modal \*/

    toggle = () => {

        this.setState({

            modal: !this.state.modal,

        });

        if (((!this.state.modal) && this.state.editId === null) || this.state.modal === true) {

            this.formReset()

        }

    }

    /\* input field change function \*/

    onInputChange = (e) => {

        this.setState({

            profile: {

                ...this.state.profile, [e.target.name]: e.target.value,

                id: (this.state.editId !== null) ? this.state.editId : Date.now()

            }

        })

    }

    /\* hobby field change function \*/

    onHobbyChange = (e) => {

        if (e.target.checked) {

            this.setState({

                profile: {

                    ...this.state.profile, hobby: [...this.state.profile.hobby, e.target.checked && e.target.value]

                }

            })

        }

        else {

            this.setState({

                profile: { ...this.state.profile, hobby: this.state.profile.hobby.filter((item) => item !== e.target.value) }

            })

        }

    }

    /\* On confirm password \*/

    confirmPassword = (e) => {

        this.setState({

            cPassword: e.target.value

        })

    }

    /\* get local storage function \*/

    getLocal = () => {

        this.setState({ allstudent: JSON.parse(localStorage.getItem("allstudent")) }, () => {

        })

    }

    /\* set local storage function \*/

    setLocal = () => {

        localStorage.setItem("allstudent", JSON.stringify(this.state.allstudent));

    }

    /\* submit function \*/

    insertRow = (e) => {

        e.preventDefault();

        if (this.state.editId === null && (this.state.profile.password !== this.state.cPassword)) {

            Swal.fire({

                imageUrl: logo,

                imageAlt: 'A tall image',

                title:'Wrong Password'

            }

            )

        }

        else {

            if (this.state.editId !== null) {

                this.toggle()

                this.state.allstudent.splice(this.state.allstudent.findIndex((item) => item.id === this.state.editId), 1, this.state.profile)

                this.setState({

                    editId: null

                })

                this.setLocal();

                this.getLocal();

            }

            else {

                this.toggle()

                this.setState({

                    allstudent:

                        [...this.state.allstudent, this.state.profile]

                }, () => {

                    this.setLocal();

                    this.getLocal();

                })

            }

        }

    }

    componentDidMount() {

        if (JSON.parse(localStorage.getItem("allstudent")) !== null) {

            this.getLocal()

        }

    }

    /\* Edit Function \*/

    editRow = (id) => {

        this.setState({ editId: id, profile: this.state.allstudent.filter((item) => item.id === id)[0] }, () => {

            this.toggle()

        })

    }

    /\* Delete Function \*/

    deleteRow = (id) => {

        if (window.confirm("Are You Sure You Want to Delete This Row")) {

            this.state.allstudent.splice(this.state.allstudent.findIndex((elment) => elment.id === id), 1)

            this.setLocal();

            this.getLocal();

            // this.formReset();

        }

    }

    /\* Form reset function \*/

    formReset = () => {

        this.setState({

            profile: {

                id: '',

                firstName: '',

                lastName: '',

                userName: '',

                password: '',

                email: '',

                age: null,

                phoneNo: null,

                gender: '',

                hobby: []

            },

        })

    }

    render() {

        return (

            <Fragment>

                <Container fluid className='bg-dark'>

                    <Container>

                        <div className='d-flex justify-content-between py-3'>

                            <h1 className='text-white'>STUDENT</h1>

                            <Button className='text-white rounded-pill px-5 fs-4' onClick={this.addFunction}>+ADD</Button>

                        </div>

                    </Container>

                </Container>

                <Modal isOpen={this.state.modal} toggle={this.toggle} className={this.props.className} backdrop={this.state.backdrop}>

                    <ModalHeader toggle={this.toggle}><p className='fw-bold fs-3'>Student Information</p></ModalHeader>

                    <Form id="form" onSubmit={this.insertRow}>

                        <ModalBody>

                            <Row className=''>

                                <Col xs={12} md={4}>

                                    <div className="mb-3">

                                        <Label for="InputOne" className='fw-bold'>First Name</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="text" id="InputOne"

                                            placeholder="First name"

                                            value={(this.state.profile.firstName) ?? ''}

                                            name="firstName" />

                                    </div>

                                    <div className="mb-3">

                                        <Label for="InputFive" className='fw-bold'>Phone No</Label>

                                        <Input

                                            type="tel" id="InputFive"

                                            placeholder="Your Mobile No"

                                            onChange={this.onInputChange}

                                            value={(this.state.profile.phoneNo) ?? ''}

                                            name="phoneNo" />

                                    </div>

                                    {

                                        (this.state.editId === null)

                                        &&

                                        <div className="mb-3">

                                            <Label for="InputSix" className='fw-bold'>Password</Label>

                                            <Input

                                                onChange={this.onInputChange}

                                                type="password"

                                                id="InputSix"

                                                placeholder="Password"

                                                name="password" />

                                        </div>

                                    }

                                    <div className="mt-2">

                                        <Label for="radioDiv" className='fw-bold'>Gender</Label>

                                        <div id="radioDiv" className="ps-2">

                                            <div className="form-check-inline">

                                                <Input

                                                    type="radio"

                                                    value="male"

                                                    id="radioOne"

                                                    name="gender"

                                                    onChange={this.onInputChange}

                                                    checked={this.state.profile.gender === 'male'}

                                                />

                                                <Label for="radioOne" className='ms-2'>Male</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    type="radio"

                                                    value="female"

                                                    id="radioTwo"

                                                    name="gender"

                                                    onChange={this.onInputChange}

                                                    checked={this.state.profile.gender === 'female'} />

                                                <Label for="radioTwo" className='ms-2'>Female</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    type="radio"

                                                    value="trans"

                                                    id="radioThree"

                                                    name="gender"

                                                    onChange={this.onInputChange}

                                                    checked={this.state.profile.gender === 'trans'} />

                                                <Label for="radioThree" className='ms-2'>Trans</Label>

                                            </div>

                                        </div>

                                    </div>

                                </Col>

                                <Col xs={12} md={4}>

                                    <div className="mb-3">

                                        <Label for="InputTwo" className='fw-bold'>Last Name</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="text"

                                            id="InputTwo"

                                            placeholder="Last name"

                                            value={(this.state.profile.lastName) ?? ''}

                                            name="lastName" />

                                    </div>

                                    <div className="mb-3">

                                        <Label for="InputThree" className='fw-bold'>Email address</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="email"

                                            id="InputThree"

                                            placeholder="Email"

                                            value={(this.state.profile.email) ?? ''}

                                            name="email" />

                                    </div>

                                    {

                                        (this.state.editId === null)

                                        &&

                                        <div className="mb-3">

                                            <Label for="InputSeven" className='fw-bold'>Confirm Password</Label>

                                            <Input

                                                onChange={this.confirmPassword}

                                                type="password" id="InputSeven"

                                                placeholder="Confirm Password"

                                                name="Cpass" />

                                        </div>

                                    }

                                    <div>

                                        <Label for="hobbyDiv" className='fw-bold'>Hobby</Label>

                                        <div id="hobbyDiv" className="ps-2">

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="cricket"

                                                    id="checkOne"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('cricket')}

                                                />

                                                <Label for="flexCheckDefault" className='ms-2'>Cricket</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="football"

                                                    id="checkTwo"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('football')} />

                                                <Label for="flexCheckDefault" className='ms-2'>Football</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="volleyball"

                                                    id="checkThree"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('volleyball')} />

                                                <Label for="checkThree" className='ms-2'>Volleyball</Label>

                                            </div>

                                            <div className="form-check-inline">

                                                <Input

                                                    className="check"

                                                    type="checkbox"

                                                    value="carom"

                                                    id="checkFour"

                                                    name="hobby"

                                                    onChange={this.onHobbyChange}

                                                    checked={this.state.profile.hobby.includes('carom')} />

                                                <Label for="checkFour" className='ms-2'>Carom</Label>

                                            </div>

                                        </div>

                                    </div>

                                </Col>

                                <Col xs={12} md={4}>

                                    <div className="mb-3">

                                        <Label for="InputNine" className='fw-bold'>User Name</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="text" id="InputNine"

                                            placeholder="User Name"

                                            value={(this.state.profile.userName) ?? ''}

                                            name="userName" />

                                    </div>

                                    <div className="mb-3">

                                        <Label for="InputFour" className='fw-bold'>Age</Label>

                                        <Input

                                            onChange={this.onInputChange}

                                            type="number"

                                            id="InputFour"

                                            placeholder="Your Age"

                                            value={(this.state.profile.age) ?? ''}

                                            name="age" />

                                    </div>

                                </Col>

                            </Row>

                        </ModalBody>

                        <ModalFooter>

                            <Button

                                type="submit"

                                className="btn-success"

                                id="submitBtn"

                                onClick={this.submitButton}>Submit</Button>

                            <Button type="button" className="btn-danger" id="cancelBtn"

                                onClick={this.toggle}>Cancel</Button>

                        </ModalFooter>

                    </Form>

                </Modal>

                <Container fluid>

                    <Table responsive striped className='border border-1 '>

                        <caption className="fw-bolder text-center caption-top text-uppercase fs-3">Student personal information

                        </caption>

                        <thead id="thead" className="bg-dark">

                            <tr>

                                <th>SR. NO.</th>

                                <th>User name</th>

                                <th>First name</th>

                                <th>Last name</th>

                                <th>Email</th>

                                <th>Age</th>

                                <th>Telephone</th>

                                <th>Gender</th>

                                <th>Hobby</th>

                                <th>Action</th>

                            </tr>

                        </thead>

                        <tbody id="tbody">

                            {

                                (this.state.allstudent !== [])

                                &&

                                this.state.allstudent.map((item, index) => {

                                    return (

                                        <tr className="tr" key={index}>

                                            <th>{index + 1}</th>

                                            <th>{item.userName}</th>

                                            <th>{item.firstName}</th>

                                            <th>{item.lastName}</th>

                                            <th>{item.email}</th>

                                            <th>{item.age}</th>

                                            <th>{item.phoneNo}</th>

                                            <th>{item.gender}</th>

                                            <th>{item.hobby.join(',')}</th>

                                            <th>

                                                <Button className='me-2' onClick={() => this.editRow(item.id)}>Edit</Button>

                                                <Button onClick={() => this.deleteRow(item.id)}>Delete</Button>

                                            </th>

                                        </tr>

                                    )

                                })

                            }

                        </tbody>

                    </Table>

                </Container>

            </Fragment >

        )

    }

}

export default Header;

**Main component CSS**

.fade {

    background-color: rgba(0, 0, 0, 0.8);

}

.modal {

    position: fixed;

    z-index: 999;

    top: 0;

    right: 0;

    left: 0;

    bottom: 0;

}

@media (max-width: 1500px) {

    .modal-dialog {

        max-width: 900px !important;

        margin-right: auto;

        margin-left: auto;

    }

}

**UseState**

const { useState, Fragment } = require("react")

const UseState = () => {

    /\* We can Declare State like this

    in t his first user name state is declare second parameter is xyz which is by deafult function generate when we make hooks. Name of this function is anything but as a statndard we give name like set with state name

    like age => setAge

         userName =>setuserName

         we can make state with initial value or without value

         like this

        =>Without value

         useState()

        =>With value

         useState(20)  =>With value

         useState([20])  =>With value

         useState({

            firstName:'Mehul',

            lastName:'Waghela'

         })  =>With value

        issue=> we can not print holl object in dom therefore we have ro access only property and print it

         \*/

    const [user, xyz] = useState()

    const [age, setAge] = useState(20)

    const [firstName, setFirstName] = useState('Mehul')

    const [number, setNumber] = useState([10, 20, 3, 46, 54, 'mehul'])

    const [object, setObject] = useState({

        firstName: "Oscar", age: 25, profile: {

            firstName: "Oscar", age: 50

        }

    })

    const [objectarr, setObjectArray] = useState({ firstName: "Oscar" })

    const changeUser = () => {

        xyz('admin')

    }

    const changeAge = () => {

        setAge(age + 5)

    }

    const changeFirstName = () => {

        setFirstName('Mehul Waghela')

    }

    const changeNumber = () => {

        setNumber([50, 60, 70])

    }

    const changeObject = () => {

        setObject({ ...object, age: 50, profile: { ...object.profile, age: 54 } })

    }

    const changeObjectinArray = () => {

        setObjectArray([10, 20, 30])

    }

    return (

        <Fragment>

            <h1>User :{user}</h1>

            <h1>Age :{age}</h1>

            <h1>Firstname :{firstName}</h1>

            <h1>Array :{number.join(',')}</h1>

            <h1>object :{object.firstName}</h1>

            <h1>object :{object.age}</h1>

            <h1>object :{object.profile.age}</h1>

            {/\* if firstname property not avalilable then output 'hello \*/}

            <h1>object :{objectarr.firstName ?? 'Hello'}</h1>

            <button onClick={changeUser}>Change User</button>

            <button onClick={changeAge}>Change Age</button>

            <button onClick={changeFirstName}>Change First Name</button>

            <button onClick={changeNumber}>Change Array</button>

            <button onClick={changeObject}>Change object</button>

            <button onClick={changeObjectinArray}>Change object</button>

        </Fragment>

    )

}

export default UseState

**Use Effect**

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

const Effect = () => {

    const [message, setMessage] = useState('Good Morning')

    const [age, setAge] = useState(50)

    /\* this use effect is with no dependency

    no dependency means aa function by default ek vaar call thase ane pachi kai pan changes thay component or state ma tyare aa function call thay \*/

    useEffect(() => {

        console.log("No Dependencies")

    })

    /\* no dependency means aa function render pela ek vaar call thase ane pachi jo aapre bija page ma jaiye aj website ma ne pachi pacha a page ma aaviye to a call thase, a sivay a biji vaar koi pan vaar call nai thay\*/

    useEffect(() => {

        console.log("Blank Dependencies")

    }, [])

    /\* with dependency means a ek vaar to by default call thase ane pachi je dependencies ma je state lakhyu ama jo changes thay to j aa function call thase bija state ma change thay to aa function call nai thay

    aapre dependencies ma only statej aapi shakiye ane a pan aj state je use state thi banya hoy

    =>multiple dependencies aapre change kari shakiye

    useEffect(() => {

        console.log("With Dependencies")

    }, [message,age])  <=== like this we can give multiple dendency

    \*/

    useEffect(() => {

        console.log("With Dependencies message")

    }, [message])

    /\* jyare age athwa message koi pan ek ma pan state ma change thay to aa function call thase \*/

    useEffect(() => {

        console.log("With Dependencies message and age")

    }, [message, age])

    return (

        <Fragment>

            <h1>{message}</h1>

            <h1>{age}</h1>

            <button onClick={() => setMessage(25)}>set message</button>

            <button onClick={() => setAge(age + 1)}>Age set</button>

        </Fragment>

    )

}

export default Effect

**Hooks Task**

import { Fragment, useState } from "react";

const Login = () => {

    const [login, setLogin] = useState(false)

    const [message, setMesssge] = useState({

        userMessage: 'Guest user',

        loginState: 'Please Login'

    })

    const changeLogin = () => {

        setLogin(!login)

        if (login !== true) {

            setMesssge({

                userMessage: 'Welcome Mehul Waghela',

                loginState: 'You are Online'

            })

        }

        else {

            setMesssge({

                userMessage: 'Guest user',

                loginState: 'Please Login'

            })

        }

    }

    return (

        <Fragment>

            <h1>{message.userMessage}</h1>

            <h2>{message.loginState}</h2>

            <button onClick={changeLogin}>{login ? 'offline' : 'online'}</button>

        </Fragment>

    )

}

export default Login

**Memo**

**Parent Component**

import './App.css';

import FunctionComponent from './components/FunctionComponent';

import { Component, Fragment } from 'react';

class App extends Component {

  constructor() {

    super()

    this.state = {

      passage: 40,

      age: 50

    }

  }

  abc = () => {

    console.log("a")

    this.setState({ age: 77 })

  }

  /\* in this class component we only change age on click on set age button and another component wrape in return then app js component render and Functional component not require rerender then we can stop rerendering using memo

  In this code we pass the data passage which is not change during click function therefore use memo to stop unnecessary rendering\*/

  render() {

    return (

      <div className="App" >

        <Fragment>

          <FunctionComponent passage={this.state.passage} />

          <h1>parent{this.state.age}</h1>

          <button onClick={this.abc}>set age</button>

        </Fragment>

      </div>

    )

  }

}

export default App;

**Child Component**

import { Fragment, useState,memo } from "react";

const FunctionComponent = ({ passage }) => {

    const [age, setAge] = useState(20)

    const changeAge = () => {

        setAge(passage)

    }

    console.log("hello")

    return (

        <Fragment>

            <h1>Good Morning</h1>

            <h1>{age}</h1>

            <button onClick={changeAge}>set age</button>

        </Fragment>

    )

}

export default memo(FunctionComponent)

**Error Boundary**

**Wrapper**

import { Component, Fragment } from 'react'

import Greet from './Greet';

import ErrorBoundries from './ErrorBoundries';

class Wrapper extends Component {

    constructor() {

        super()

        this.state = {

            main: false

        }

    }

    changeState = () => {

        this.setState({ main: !this.state.main })

    }

    /\* if in the program any one error occur our website crash and all process stop therefore it is big problem. This problem solved by Error Boundries. Error boundries we can not test because when we run project if error occur then it is give error and program stop run therefore we can not test error boundary and when we work with API if object format is not proper then it is give error.In this program I make error on perticular condition in greet component and it is wrap in Error Boundary Component.

Error Boundary has two method which is call when error occur

method 1=> getDerivedStateFromError(error){

}

method 2=> componentDidCatch(error, info) {

        console.log("Error", error);

        console.log("Info", info)

    }

}

    Aa method Error aave tyare call thase ane jo error aave to a condition a error part ne return karvana badle particular message return karavi shakiye\*/

    /\* ahiya jo error hoy to hasError ne true karyo ane hasError true ni condition par message return karyo Oops ane jo erro nai hoy to a component return karyo(Greet) je as a HTML element pass karyo che  means error aave to pan bija codes run thase ane output pan dekhase ane website crash nai thay \*/

    // componentDidCatch(error, info) {

    //     console.log("Error", error)

    //     console.log("Info", info)

    //     return { hasError: true }

    // }

    render() {

        return (

            <Fragment Fragment >

                <button onClick={this.changeState}>Change State</button>

                {

                    ((this.state.main === false) &&

                        <div>

                            <ErrorBoundries>

                                <Greet message={'Oscar'} />

                            </ErrorBoundries>

                            <ErrorBoundries>

                                <Greet message={'Career'} />

                            </ErrorBoundries>

                            <ErrorBoundries>

                                <Greet message={'Point'} />

                            </ErrorBoundries>

                        </div>

                    )

                }

                {

                    ((this.state.main === true) &&

                        <div>

                            <ErrorBoundries>

                                <Greet message={'Oscar'} />

                                <Greet message={'Career'} />

                                <Greet message={'Point'} />

                            </ErrorBoundries>

                        </div>

                    )

                }

            </Fragment >

        )

    }

}

export default Wrapper;

**ErrorBoundry Component**

import { Component } from 'react'

class ErrorBoundries extends Component {

    constructor(props) {

        super(props)

        this.state = {

            hasError: false

        }

    }

    static getDerivedStateFromError(error) {

        return { hasError: true }

    }

    render() {

        return this.state.hasError ? <h3>Oops!!! Something Went Wrong.</h3> : this.props.children

    }

}

export default ErrorBoundries

**Greet Component**

import React from 'react'

const Greet = ({ message }) => {

    if (message === 'Point') {

        throw new Error("Point is not a valid message")

    }

    else {

        return (

            <h1>{message}</h1>

        )

    }

}

export default Greet

Bootstrap

import React from 'react'

/\* run Bootstrap Steps

1=>First Install Bootstrap using =>npm install --save bootstrap

2=>Put this link in app.js file => import 'bootstrap/dist/css/bootstrap.min.css';

3=>All Class replace with className

\*/

function Style() {

    return (

        <div className="accordion" id="accordionExample">

            <div className="accordion-item">

                <h2 className="accordion-header">

                    <button className="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">

                        Accordion Item #1

                    </button>

                </h2>

                <div id="collapseOne" className="accordion-collapse collapse show" data-bs-parent="#accordionExample">

                    <div className="accordion-body">

                        <strong>This is the first item's accordion body.</strong> It is shown by default, until the collapse plugin adds the appropriate classNamees that we use to style each element. These classNamees control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.

                    </div>

                </div>

            </div>

            <div className="accordion-item">

                <h2 className="accordion-header">

                    <button className="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">

                        Accordion Item #2

                    </button>

                </h2>

                <div id="collapseTwo" className="accordion-collapse collapse" data-bs-parent="#accordionExample">

                    <div className="accordion-body">

                        <strong>This is the second item's accordion body.</strong> It is hidden by default, until the collapse plugin adds the appropriate classNamees that we use to style each element. These classNamees control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.

                    </div>

                </div>

            </div>

            <div className="accordion-item">

                <h2 className="accordion-header">

                    <button className="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseThree" aria-expanded="false" aria-controls="collapseThree">

                        Accordion Item #3

                    </button>

                </h2>

                <div id="collapseThree" className="accordion-collapse collapse" data-bs-parent="#accordionExample">

                    <div className="accordion-body">

                        <strong>This is the third item's accordion body.</strong> It is hidden by default, until the collapse plugin adds the appropriate classNamees that we use to style each element. These classNamees control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.

                    </div>

                </div>

            </div>

        </div>

    )

}

export default Style

App.js

import 'bootstrap/dist/css/bootstrap.min.css';

import './App.css';

import Style from './components/Style';

function App() {

  return (

    <div>

      <Style />

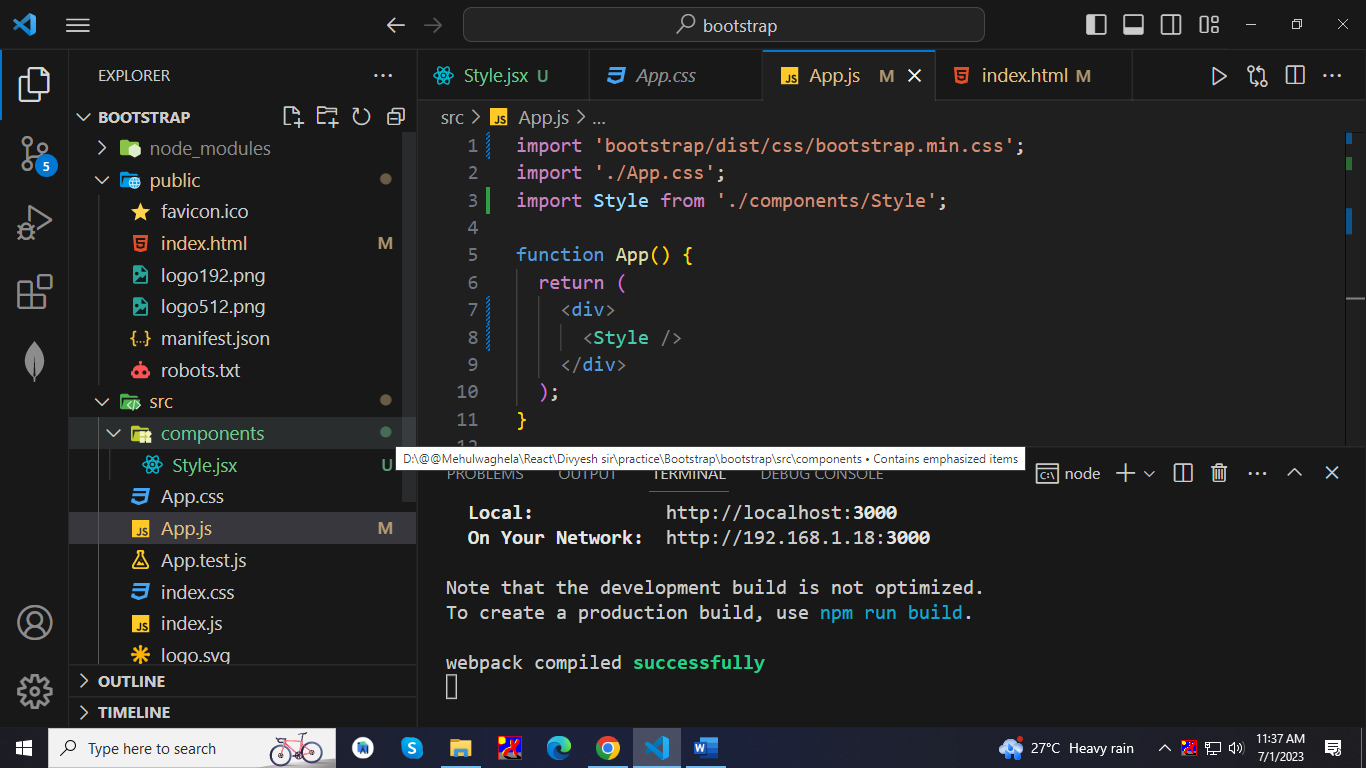
    </div>

  );

}

export default App;

Folder Structure



React Bootstrap

import React from 'react'

import { Accordion } from 'react-bootstrap'

/\* React- bootstrap Website =>https://react-bootstrap.netlify.app/docs/getting-started/introduction

1=> First install bootstrap and react-bootstrap using command

   npm install react-bootstrap bootstrap

2=>Then check in dependency that install or not

3=>Then import Bootstrap in app.js => import 'bootstrap/dist/css/bootstrap.min.css';

   In react-bootstrap we not require script.js file of bootstrap\*/

function Style() {

    return (

        <div>

            <Accordion defaultActiveKey="0">

                <Accordion.Item eventKey="0">

                    <Accordion.Header>Accordion Item #1</Accordion.Header>

                    <Accordion.Body>

                        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do

                        eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad

                        minim veniam, quis nostrud exercitation ullamco laboris nisi ut

                        aliquip ex ea commodo consequat. Duis aute irure dolor in

                        reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla

                        pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

                        culpa qui officia deserunt mollit anim id est laborum.

                    </Accordion.Body>

                </Accordion.Item>

                <Accordion.Item eventKey="1">

                    <Accordion.Header>Accordion Item #2</Accordion.Header>

                    <Accordion.Body>

                        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do

                        eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad

                        minim veniam, quis nostrud exercitation ullamco laboris nisi ut

                        aliquip ex ea commodo consequat. Duis aute irure dolor in

                        reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla

                        pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

                        culpa qui officia deserunt mollit anim id est laborum.

                    </Accordion.Body>

                </Accordion.Item>

            </Accordion>

        </div>

    )

}

export default Style

App.js

import 'bootstrap/dist/css/bootstrap.min.css';

import './App.css';

import Style from './Components/Style';

function App() {

  return (

    <div className="App">

      <Style/>

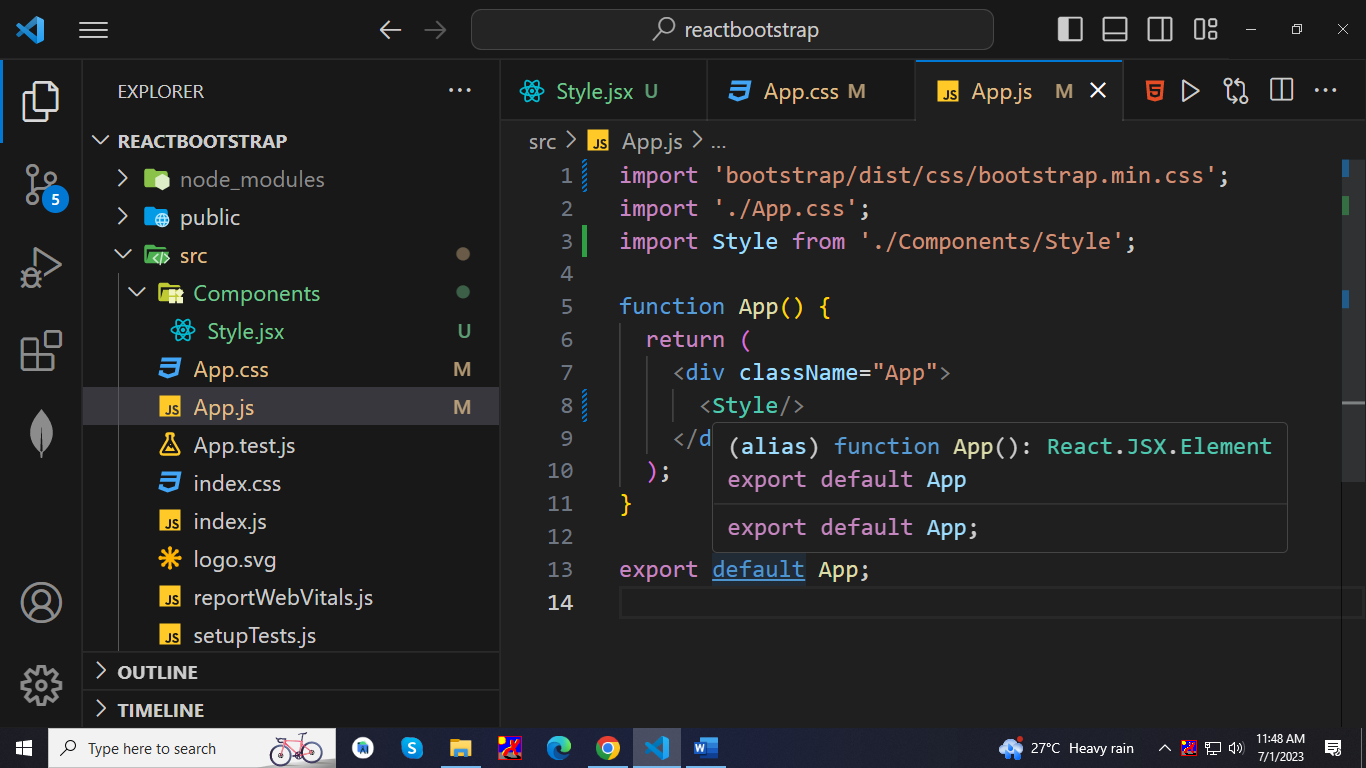
    </div>

  );

}

export default App;

Folder Structure



Listing

App.js

import './App.css';

import 'bootstrap/dist/css/bootstrap.min.css'

import CardComponent from './components/CardComponent';

import CardClass from './components/CardClass';

import img1 from './assets/image/pexels-adrienn.jpg'

import img2 from './assets/image/pexels-apg.jpg'

import img3 from './assets/image/pexels-bhoopal.jpg'

import img4 from './assets/image/pexels-gianne.jpg'

import img5 from './assets/image/pexels-phil.jpg'

import img6 from './assets/image/pexels-pixabay.jpg'

import img7 from './assets/image/pexels-tuấn-kiệt.jpg'

import img8 from './assets/image/pexels-tuấn.jpg'

import img9 from './assets/image/pixabay.jpg'

import img10 from './assets/image/tuấn-kiệt-jr.jpg'

function App() {

  return (

    <div className="App">

      <div style={{ display: 'flex' }}>

        <CardComponent title="Mehul" sub="Waghela" image={img1} />

        <CardComponent title="Mehul" sub="Waghela" image={img2} />

        <CardComponent title="Mehul" sub="Waghela" image={img3} />

        <CardComponent title="Mehul" sub="Waghela" image={img4} />

        <CardComponent title="Mehul" sub="Waghela" image={img5} />

        <CardComponent title="Mehul" sub="Waghela" image={img6} />

        <CardComponent title="Mehul" sub="Waghela" image={img7} />

        <CardComponent title="Mehul" sub="Waghela" image={img8} />

        <CardComponent title="Mehul" sub="Waghela" image={img9} />

        <CardComponent title="Mehul" sub="Waghela" image={img10} />

      </div>

      <div style={{ display: 'flex' }}>

        <CardClass title="Mehul" sub="Waghela" image={img1} />

        <CardClass title="Mehul" sub="Waghela" image={img2} />

        <CardClass title="Mehul" sub="Waghela" image={img3} />

        <CardClass title="Mehul" sub="Waghela" image={img4} />

        <CardClass title="Mehul" sub="Waghela" image={img5} />

        <CardClass title="Mehul" sub="Waghela" image={img6} />

        <CardClass title="Mehul" sub="Waghela" image={img7} />

        <CardClass title="Mehul" sub="Waghela" image={img8} />

        <CardClass title="Mehul" sub="Waghela" image={img9} />

        <CardClass title="Mehul" sub="Waghela" image={img10} />

      </div>

    </div>

  );

}

export default App;

Function card

import 'bootstrap/dist/css/bootstrap.min.css'

import { Card, CardBody, CardTitle, CardSubtitle, CardText, Button } from 'reactstrap'

const CardComponent = (props) => {

    return (

        <div>

            <Card Card

                color="warning"

                inverse

                style={{

                    width: '18rem'

                }}

            >

                <img

                    alt="Sample"

                    src={props.image}

                    style={{ objectFit: 'cover', width: "100%", height: '250px' }}

                />

                <CardBody>

                    <CardTitle tag="h5">

                        {props.title}

                    </CardTitle>

                    <CardSubtitle

                        className="mb-2 text-muted"

                        tag="h6"

                    >

                        {props.sub}

                    </CardSubtitle>

                    <CardText>

                        Some quick example text to build on the card title and make up the bulk of the card‘s content.

                    </CardText>

                    <Button>

                        Button

                    </Button>

                </CardBody>

            </Card >

        </div>

    )

}

export default CardComponent

Class Card

import { Component } from 'react'

import { Card, CardBody, CardTitle, CardSubtitle, CardText, Button } from 'reactstrap'

class CardClass extends Component {

    render() {

        return (

            <div>

                <Card Card

                    color="warning"

                    inverse

                    style={{

                        width: '18rem'

                    }}

                >

                    <img

                        alt="Sample"

                        src={this.props.image}

                        style={{ objectFit: 'cover', width: "100%", height: '250px' }}

                    />

                    <CardBody>

                        <CardTitle tag="h5">

                            {this.props.title}

                        </CardTitle>

                        <CardSubtitle

                            className="mb-2 text-muted"

                            tag="h6"

                        >

                            {this.props.sub}

                        </CardSubtitle>

                        <CardText>

                            Some quick example text to build on the card title and make up the bulk of the card‘s content.

                        </CardText>

                        <Button>

                            Button

                        </Button>

                    </CardBody>

                </Card >

            </div>

        )

    }

}

export default CardClass

React Styling

Css.jsx

import '../style/external.css'

import '../style/abc.scss'

import ButtonModule from '../style/button.module.css'

const InlineCss = () => {

    return (

        <h1 style={{ backgroundColor: "yellow" }}>Inline style</h1>

    )

}

const Stylesheet = () => {

    let abc = { backgroundColor: "red" }

    return (

        <h1 style={abc}>stylesheet or inline style</h1>

    )

}

const External = () => {

    return (

        <h1 className="main">External style</h1>

    )

}

const ModuleCSS = () => {

    return (

        <h1 className={ButtonModule.btn}>Module style</h1>

    )

}

const SassCSS = () => {

    return (

        <div className="sub">

            <h1 className='sub\_h1'>Saas style</h1>

            <h1 className='subAbc'>Saas style</h1>

        </div>

    )

}

export { InlineCss, Stylesheet, External, ModuleCSS, SassCSS }

App.js

import './App.css';

import { InlineCss, Stylesheet, External, ModuleCSS, SassCSS } from './components/Css';

function App() {

  return (

    <div className="App">

      <InlineCss />

      <Stylesheet />

      <External />

      <ModuleCSS />

      <SassCSS />

    </div>

  );

}

export default App;

scss.scss

.sub {

    background-color: chartreuse;

    /\* & \_h1 means a upar lakhela object nu name check karse je hase teni sathe \_h1 join kari dese

    jena karane class banse .sub\_h1 , te class ne je style aapi hoy a lagi jase \*/

    &\_h1 {

        color: red;

        font-weight: 900;

    }

    &Abc {

        background-color: pink;

        color: blue;

    }

}

Button.module.css

.btn{

    background-color: aqua;

}

External.css

.main{

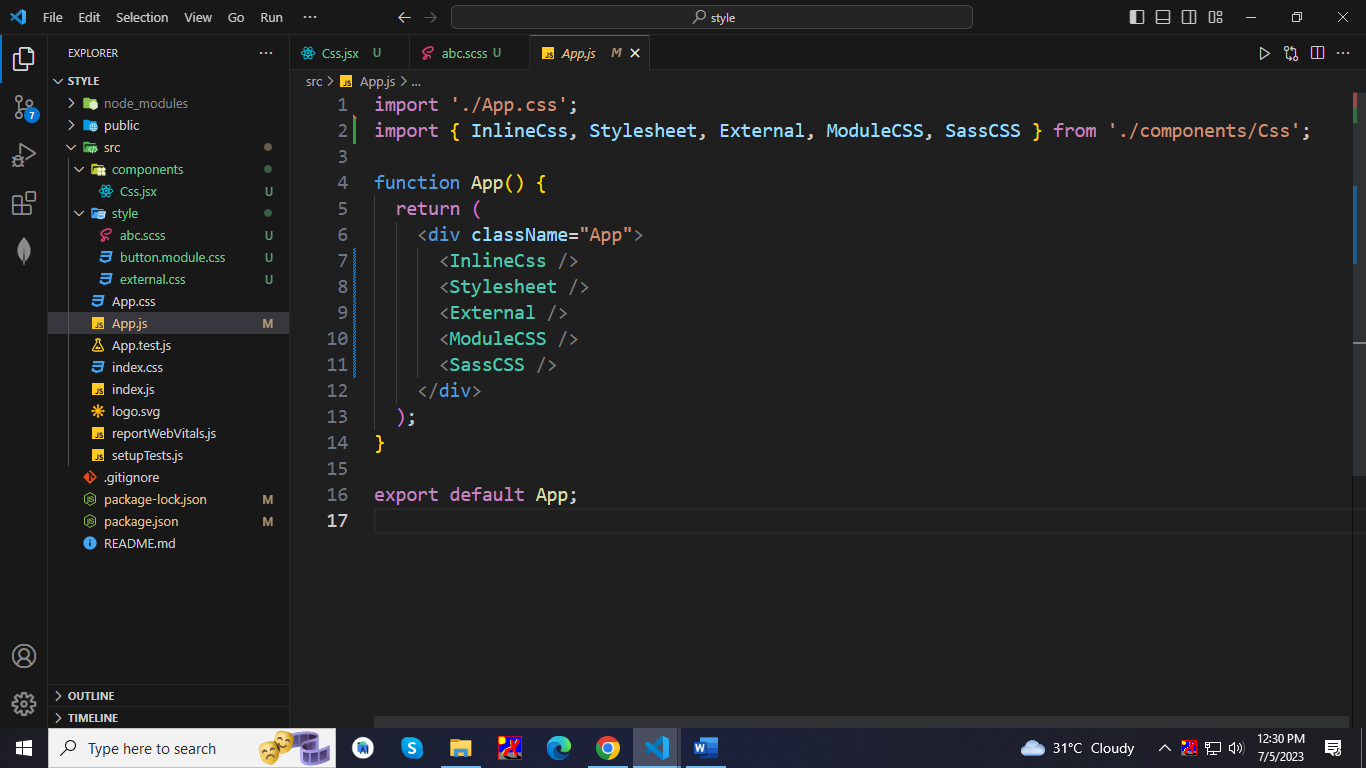
    font-size: 50px;

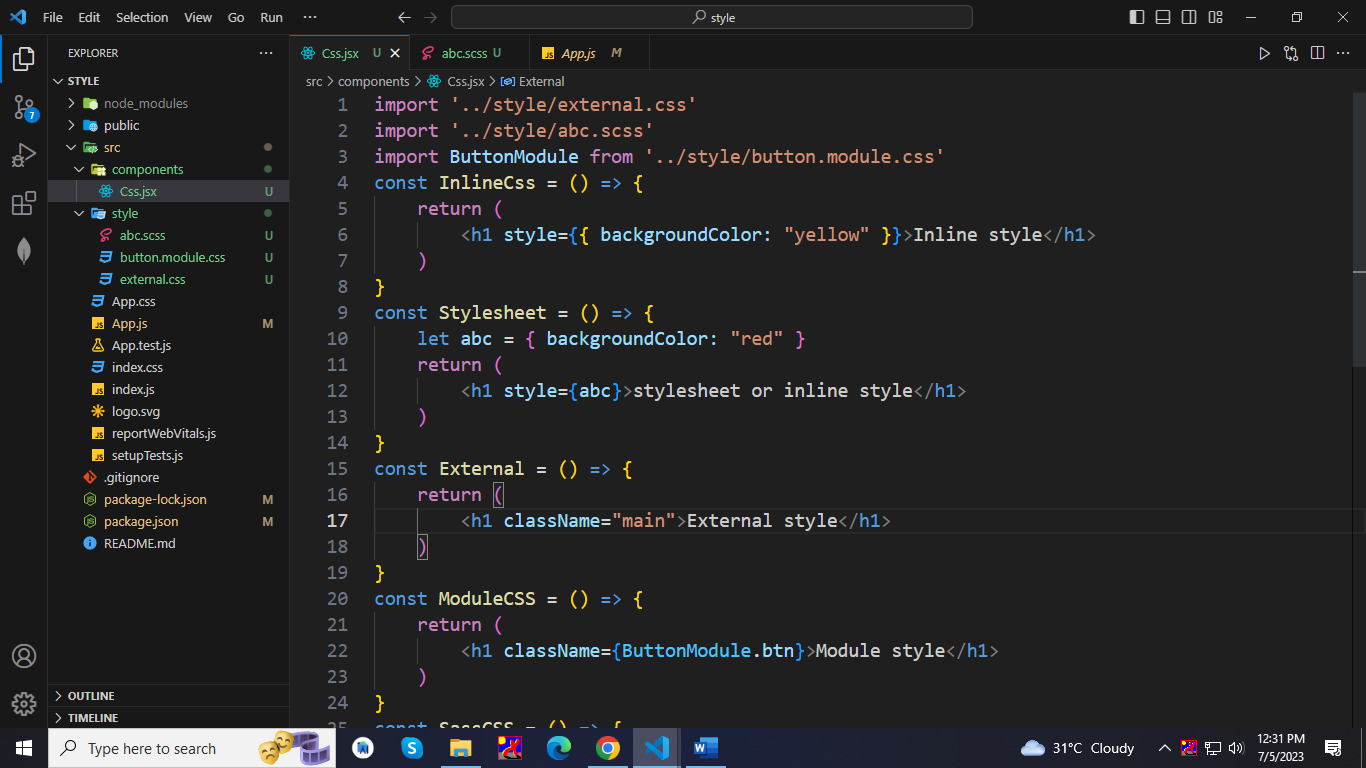
    background-color: pink;

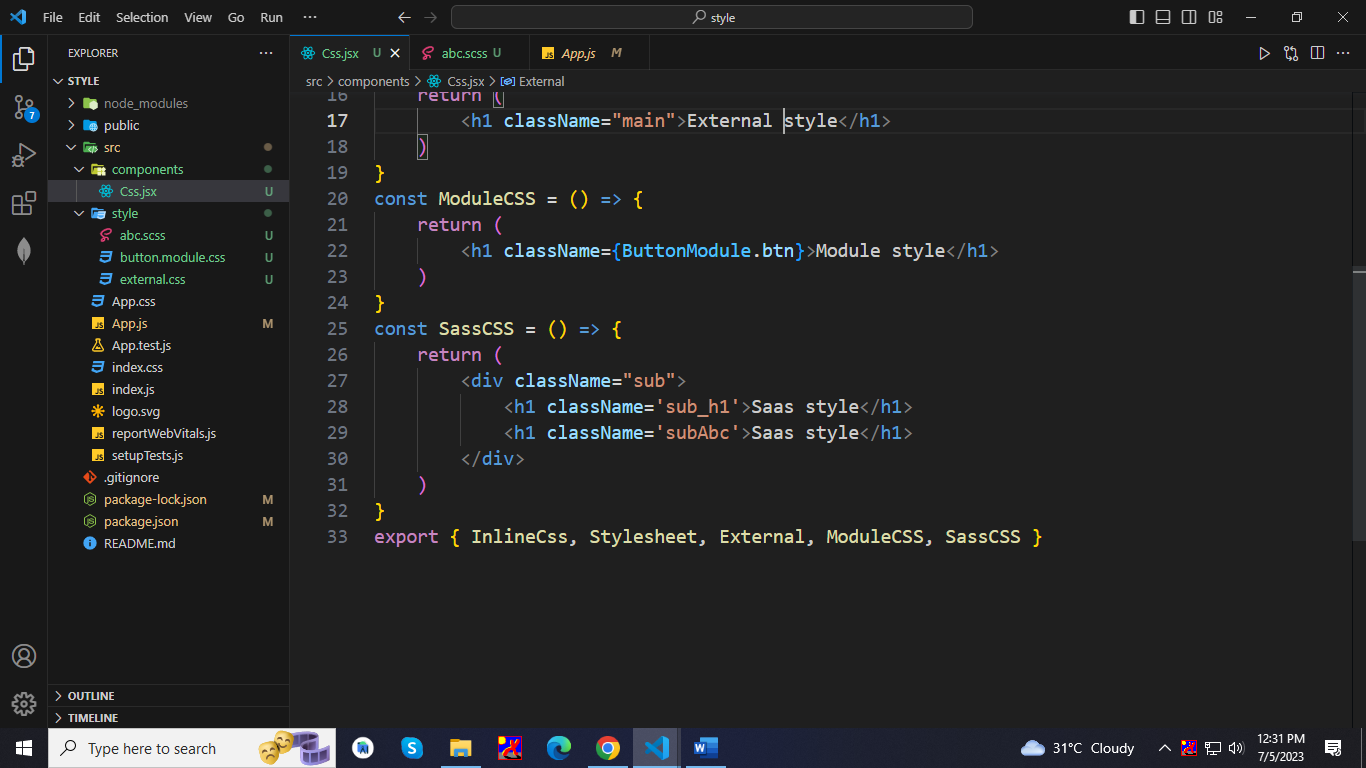
    border: 5px solid blue;

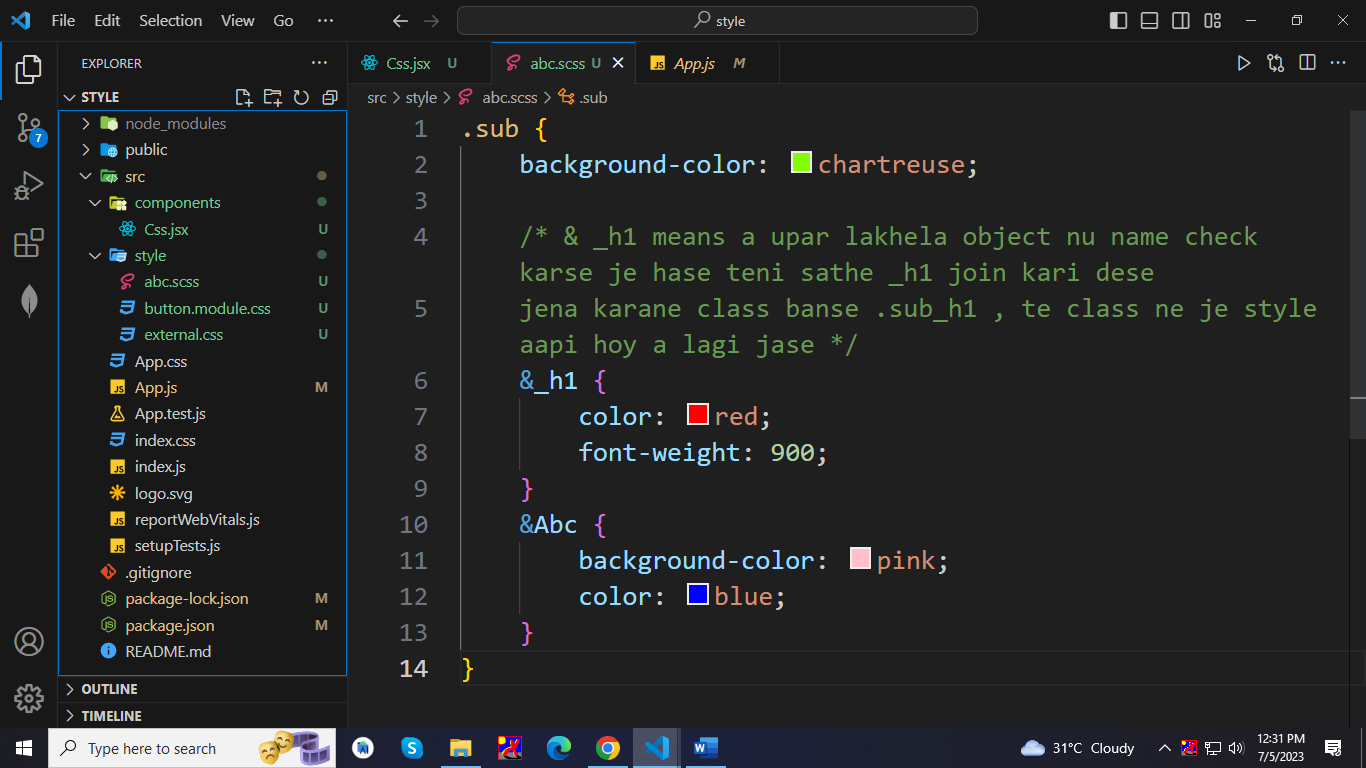
}

Folder









React-strap

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

import {

    Accordion,

    AccordionBody,

    AccordionHeader,

    AccordionItem,

} from 'reactstrap';

/\* To run react strap

1=>First install react strap =>npm install reactstrap react react-dom \*/

let arr = [1, 2, 3, 4, 56, 6]

function Style(props) {

    const [open, setOpen] = useState('1');

    const toggle = (id) => {

        if (open === id) {

            setOpen();

        } else {

            setOpen(id);

        }

    };

    return (

        <div>

            <Accordion open={open} toggle={toggle}>

                {

                    arr.map((x, i) => {

                        return (

                            <Fragment>

                                <AccordionItem>

                                    <AccordionHeader targetId={i + 1}>Accordion Item {i + 1}</AccordionHeader>

                                    <AccordionBody accordionId={i + 1}>

                                        <strong>This is the first item&#39;s accordion body.</strong>

                                        You can modify any of this with custom CSS or overriding our default

                                        variables. It&#39;s also worth noting that just about any HTML can

                                        go within the <code>.accordion-body</code>, though the transition

                                        does limit overflow.

                                    </AccordionBody>

                                </AccordionItem>

                            </Fragment>

                        )

                    })

                }

            </Accordion>

        </div>

    )

}

export default Style

App.js

import 'bootstrap/dist/css/bootstrap.min.css';

import './App.css';

import Style from './components/Style';

function App() {

  return (

    <div className="App">

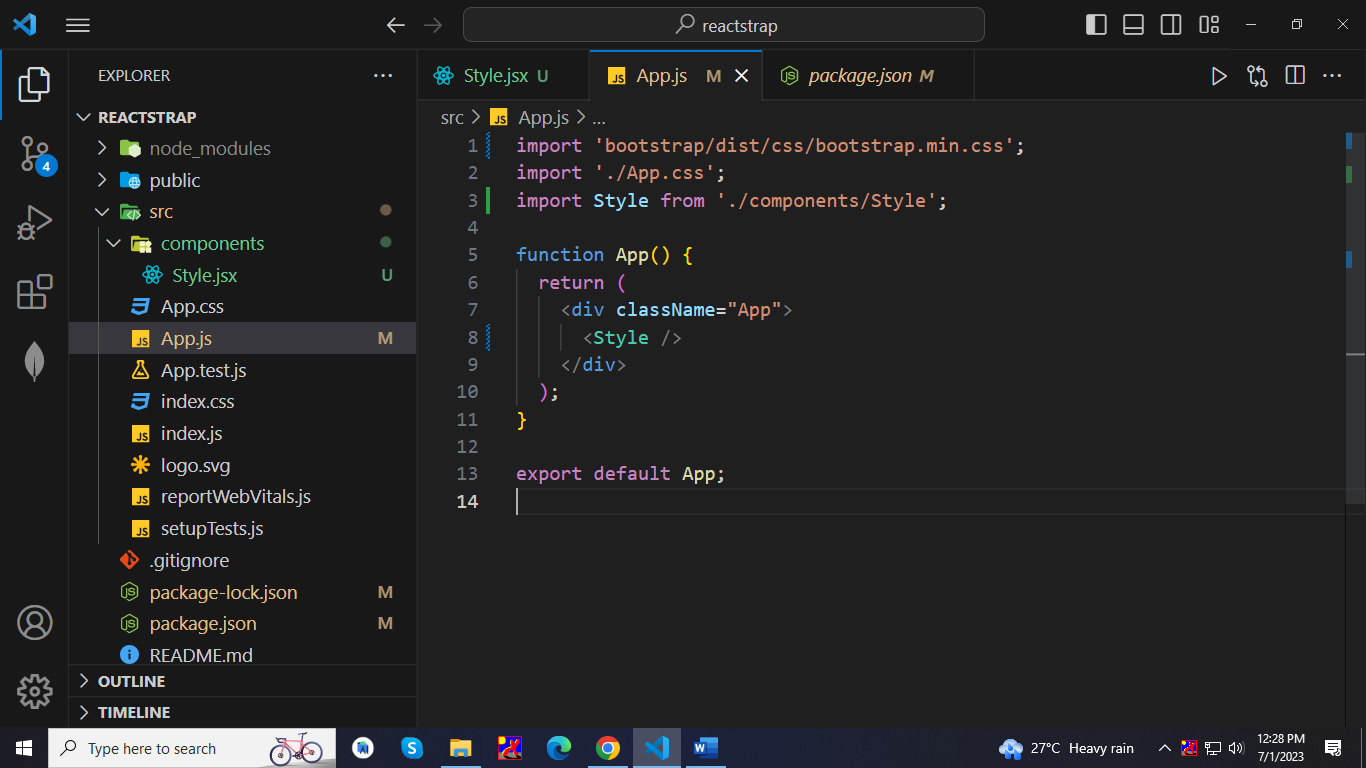
      <Style />

    </div>

  );

}

export default App;



Bootstrap with listing

import React from 'react'

/\* run Bootstrap Steps

1=>First Install Bootstrap using =>npm install --save bootstrap

2=>Put this link in app.js file => import 'bootstrap/dist/css/bootstrap.min.css';

3=>All Class replace with className

\*/

let arr = [

    { title: 'Accordion Item #1' },

    { title: 'Accordion Item #2' },

    { title: 'Accordion Item #3' },

    { title: 'Accordion Item #4' },

    { title: 'Accordion Item #5' },

]

function Style() {

    return (

        <div className="accordion" id="accordionExample">

            {arr.map((x, i) => {

                return (

                    <div className="accordion-item">

                        <h2 className="accordion-header">

                            <button className="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target={`#${i}`} aria-expanded="true" aria-controls="collapseOne">

                                {x.title}

                            </button>

                        </h2>

                        <div id={i} className="accordion-collapse collapse show" data-bs-parent="#accordionExample">

                            <div className="accordion-body">

                                <strong>This is the first item's accordion body.</strong> It is shown by default, until the collapse plugin adds the appropriate classNamees that we use to style each element. These classNamees control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.

                            </div>

                        </div>

                    </div>

                )

            })}

        </div>

    )

}

export default Style

**ADVANCE JS CRUDE**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

</head>

<body>

    <div class="container bg-body-secondary">

        <form class="row g-3 my-5" id="form">

            <div class="col-md-6">

                <label for="name" class="form-label">Name</label>

                <input type="text" class="form-control input" id="name" name="name">

            </div>

            <div class="col-md-6">

                <label for="email" class="form-label">Email</label>

                <input type="email" class="form-control input" id="email" name="email">

            </div>

            <div class="col-md-6">

                <label for="phone" class="form-label">Phone No</label>

                <input type="tel" class="form-control input" id="phone" name="phone">

            </div>

            <div class="col-md-6">

                <label for="pincode" class="form-label">Pincode</label>

                <input type="number" class="form-control input" id="pincode" name="pincode">

            </div>

            <div class="col-6">

                <label class="form-check-label" for="flexRadioDefault1">Gender</label>

                <div class="form-check col-6">

                    <input class="form-check-input" type="radio" name="gender" id="flexRadioDefault1" value="male">

                    <label class="form-check-label" for="flexRadioDefault1" name="gender">Male</label>

                </div>

                <div class="form-check col-6">

                    <input class="form-check-input" type="radio" name="gender" id="flexRadioDefault2" value="female">

                    <label class="form-check-label" for="flexRadioDefault2">

                        Female

                    </label>

                </div>

            </div>

            <div class="form-check col-6">

                <label class="form-check-label" for="flexCheckChecked">

                    languages

                </label>

                <div class="form-check">

                    <label class="form-check-label" for="flexCheckChecked">

                        Node Js

                    </label>

                    <input class="form-check-input" type="checkbox" value="node" name="language">

                </div>

                <div class="form-check">

                    <label class="form-check-label" for="flexCheckChecked">

                        Angular

                    </label>

                    <input class="form-check-input" type="checkbox" value="angualar" name="language">

                </div>

                <div class="form-check">

                    <label class="form-check-label" for="flexCheckChecked">

                        React

                    </label>

                    <input class="form-check-input" type="checkbox" value="react" name="language">

                </div>

            </div>

            <div class="col-6">

                <input type="file" name="profile" oninput="setProfile(this)">

                <img src="" alt="imageFile" id="outImage" width="150" height="150">

            </div>

            <div class="col-12 text-center">

                <button id="save" type="button" onclick="submitFunction()">Submit</button>

            </div>

        </form>

    </div>

    <div class="container my-5">

        <table class="table w-100">

            <thead>

                <tr>

                    <th scope="col">Id</th>

                    <th scope="col">Profile</th>

                    <th scope="col">Name</th>

                    <th scope="col">Email</th>

                    <th scope="col">Phone</th>

                    <th scope="col">Pin code</th>

                    <th scope="col">Gender</th>

                    <th scope="col">Language</th>

                    <th scope="col">Action</th>

                </tr>

            </thead>

            <tbody id="tbody">

            </tbody>

        </table>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"

        crossorigin="anonymous"></script>

    <script src="index.js"></script>

</body>

</html>

**JS**

let array = JSON.parse(localStorage.getItem('array')) || [] /\* jo localstorage ma data null hoy to or niconditon fullfill tahse ane array ni value blank set thase[] \*/

let count = JSON.parse(localStorage.getItem('count')) || 0

let editId = 0

let a;

document.getElementById('outImage').style.display = 'none'

const setProfile = async (e) => {

    a = await image1(e.files[0])

    document.getElementById('outImage').src = a

    document.getElementById('outImage').style.display = 'block'

}

const submitFunction = () => {

    let obj = {}

    document.querySelectorAll('.input')?.forEach((element) => {

        obj[element.name] = element.value

    })

    obj.gender = document.querySelector('input[type="radio"]:checked')?.value

    let language = []

    document.querySelectorAll('input[type="checkbox"]:checked').forEach((element) => {

        language.push(element.value)

    })

    obj.language = language

    obj.profile = a

    if (editId == 0) {

        count++

        localStorage.setItem('count', JSON.stringify(count))

        obj.id = count

        array.push(obj)

    }

    else {

        obj.id = editId

        objFind = array.findIndex(element => element.id == editId)

        array.splice(objFind, 1, obj)

        editId = 0

    }

    localStorage.setItem('array', JSON.stringify(array))

    document.getElementById('form').reset()

    maketable()

    document.getElementById('outImage').style.display = 'none'

}

maketable = () => {

    let html = ''

    array.forEach(element => {

        let str = ` <tr>

        <td>${element.id}</td>

        <td><img src=${element.profile} width='50' height='50' class='text-center'/></td>

        <td>${element.name}</td>

        <td>${element.email}</td>

        <td>${element.phone}</td>

        <td>${element.pincode}</td>

        <td>${element.gender}</td>

        <td>${element.language}</td>

        <td>

            <button onclick="edit(${element.id})">edit</button>

            <button onclick="del(${element.id})">delet</button>

        </td>

    </tr>`

        html += str

    })

    document.getElementById('tbody').innerHTML = html

}

/\* this function is use for read file any type image,video.audio and then convert

In string format if any error occur then return null else send that string data where this image1 function call and store in object when we want image back that time this dtringify data set in img tag src attribute \*/

const image1 = (file) => new Promise((resolve, reject) => {

    const reader = new FileReader();

    reader.readAsDataURL(file)

    reader.onload = () => resolve(reader.result)

    reader.onerror = () => reject

})

const del = (id) => {

    array.splice(array.findIndex(element => element.id === id), 1)

    localStorage.setItem('array', JSON.stringify(array))

    maketable()

}

const edit = (id) => {

    editId = id

    editObj = array.find(element => element.id === id)

    for (let x in editObj) {

        if (x === 'id') {

        }

        else if (x === 'profile') {

            document.getElementById('outImage').src = editObj[x]

            document.getElementById('outImage').style.display = "block"

        }

        else if (x === 'language') {

            editObj.language.forEach(element => {

                document.querySelector(`input[value=${element}]`).checked = true

            })

        }

        else if (x === 'gender') {

            ((document.querySelector(`input[value=${editObj[x]}]`))).checked = true

        }

        else {

            document.querySelector(`input[name=${x}]`).value = editObj[x]

        }

    }

}

maketable()

**Class State Management**

import { Component, Fragment } from 'react'

class ClassComponent extends Component {

    /\* in this class constructor run only one time therefore if we set any state without setstate function then it is change only construction function.

    if we want it in render then we have to use setstate function. when we change data using setstate function then it rerender the render function and data show in html

    therefore we use set interval which call setstate function on particular time and rerender done after particular time  \*/

    constructor() {

        super()

        this.state = {

            number: 123,

            arr: [100]

        }

        setInterval(() => {

            this.state.arr.push("oscar")

            console.log(this.state.arr)

            this.setState({

                number:

                    this.state.number

            })

        }, 3000);

    }

    render() {

        return (

            <Fragment>

                <h1>{this.state.number}</h1>

                <h1>{this.state.arr}</h1>

                <h1>Hello</h1>

            </Fragment>

        )

    }

}

export default ClassComponent

**Function StateManegment**

import { Fragment, useState } from "react"

const FunctionComponent = () => {

    let [count, setCount] = useState({ a: 70 })

    let [number, setNumber] = useState(50)

    const [array, setarray] = useState([10, 20, 30])

    setTimeout(() => {

        count.a = count.a + 1

        /\* When we change any state without set function then it is change but not render and if we set state using set function then it is change but it is not show below that code , it is show directly in return or Dom therefore we first data by variable name and then console show that show output immidiately and then set state using set function it rerender full code from top to bottom and data show in HTML also \*/

        count = {

            ...count

        }

        array.push(50)

        console.log(array)

        setarray([...array])

        number += 1

        setNumber(number)

    }, 2000)

    return (

        <Fragment>

            <h1>Function Component</h1>

            <h1>{array}</h1>

            <h1>{count.a}</h1>

            <h1>{number}</h1>

        </Fragment>

    )

}

export default FunctionComponent

**Function Crud**

**Main form component**

import { Fragment, useEffect, useState } from "react"

import { Col, Container, Form, FormGroup, Input, Label, Row } from "reactstrap"

import MainTable from "./MainTable"

const MainForm = () => {

    let [count, setcount] = useState(0)

    let [editid, seteditid] = useState(0)

    let [obj, setobj] = useState({

        language: []

    })

    let [editobj, seteditobj] = useState({})

    let [array, setarray] = useState([])

    const mainData = async (e) => {

        if (e.target.name === 'language') {

            if (e.target.checked) {

                obj[e.target.name] = [...obj.language, e.target.value]

            }

            else {

                // obj.language.splice(obj.language.findIndex((x) => x === e.target.value), 1)

                obj.language = obj.language?.filter((x, i) => x !== e.target.value)

            }

        }

        else if (e.target.name === 'profile') {

            obj[e.target.name] = await toBase64(e.target.files[0])

        }

        else {

            obj[e.target.name] = e.target.value

        }

        obj['id'] = count

        setobj({ ...obj })

    }

    const submitFunction = (e) => {

        e.preventDefault()

        if (editid === 0) {

            count++

            obj['id'] = count

            setobj({ ...obj })

            setcount(count)

            array.push(obj)

        }

        else {

            obj.id = editid

            setobj({ ...obj })

            array.splice(array.findIndex((x) => x.id === editid), 1, obj)

            editid = 0

        }

        setarray([...array])

        localStorage.setItem('array', JSON.stringify(array))

        localStorage.setItem('count', count)

        obj = {

            username: '',

            email: '',

            phoneno: '',

            pincode: '',

            gender: '',

            language: []

        }

        setobj({ ...obj })

    }

    function editfun(id) {

        editobj = array.find(x => x.id === id)

        obj = editobj

        editid = obj.id

        seteditid(editid)

        setobj({ ...editobj })

    }

    function deletefun(id) {

        array.splice(array.findIndex((x) => x.id === id), 1)

        setarray([...array])

        localStorage.setItem('array', JSON.stringify(array))

    }

    const toBase64 = (file) => new Promise((resolve, reject) => {

        const reader = new FileReader();

        reader.readAsDataURL(file)

        reader.onload = () => resolve(reader.result)

        reader.onerror = () => reject

    })

    useEffect(() => {

        setarray(JSON.parse(localStorage.getItem('array')))

        setcount(JSON.parse(localStorage.getItem('count')))

    }, [])

    return (

        <Fragment>

            <Container className="mt-5 p-5 border border-1 border-black rounded-2">

                <h1 className="text-center py-3">Employee Form</h1>

                <Form onSubmit={(e) => { submitFunction(e) }}>

                    <Row>

                        <Col md={6}>

                            <FormGroup>

                                <Label for="username" className="fw-600 fs-5">

                                    User Name

                                </Label>

                                <Input

                                    id="username"

                                    name="username"

                                    placeholder=""

                                    type="text"

                                    className="main"

                                    value={obj.username}

                                    onChange={mainData}

                                />

                            </FormGroup>

                        </Col>

                        <Col md={6}>

                            <FormGroup>

                                <Label for="email" className="fw-600 fs-5">

                                    Email

                                </Label>

                                <Input

                                    id="email"

                                    name="email"

                                    placeholder=""

                                    type="email"

                                    value={obj.email}

                                    className="main"

                                    onChange={mainData}

                                />

                            </FormGroup>

                        </Col>

                        <Col md={6}>

                            <FormGroup>

                                <Label for="pincode" className="fw-600 fs-5">

                                    Pincode

                                </Label>

                                <Input

                                    id="pincode"

                                    name="pincode"

                                    placeholder=""

                                    type="number"

                                    value={obj.pincode}

                                    className="main"

                                    onChange={mainData}

                                />

                            </FormGroup>

                        </Col>

                        <Col md={6}>

                            <FormGroup>

                                <Label for="phoneno" className="fw-600 fs-5">

                                    Number

                                </Label>

                                <Input

                                    id="phoneno"

                                    name="phoneno"

                                    placeholder=""

                                    value={obj.phoneno}

                                    type="number"

                                    className="main"

                                    onChange={mainData}

                                />

                            </FormGroup>

                        </Col>

                    </Row>

                    <Row>

                        <Col md={6} className="">

                            <Label

                                check

                                for="example"

                                className="fw-600 fs-5

                                my-2"

                            >

                                Language

                            </Label>

                            <div className="d-flex justify-content-start">

                                <div>

                                    <Input

                                        id="exampleCheck"

                                        name="language"

                                        type="checkbox"

                                        className="language me-2"

                                        value="html"

                                        onChange={mainData}

                                        checked={obj.language.includes('html')}

                                    />

                                    <Label

                                        check

                                        for="exampleCheck"

                                        className="px-2"

                                    >

                                        HTML

                                    </Label>

                                </div>

                                <div>

                                    <Input

                                        id="exampleCheck1"

                                        name="language"

                                        type="checkbox"

                                        className="language me-2"

                                        value="css"

                                        onChange={mainData}

                                        checked={obj.language.includes('css')}

                                    />

                                    <Label

                                        check

                                        for="exampleCheck1"

                                        className="px-2"

                                    >

                                        CSS

                                    </Label>

                                </div>

                                <div>

                                    <Input

                                        id="exampleCheck2"

                                        name="language"

                                        type="checkbox"

                                        className="language me-2"

                                        value="javascript"

                                        onChange={mainData}

                                        checked={obj.language.includes('javascript')}

                                    />

                                    <Label

                                        check

                                        for="exampleCheck2"

                                        className="px-2"

                                    >

                                        JAVASCRIPT

                                    </Label>

                                </div>

                                <div>

                                    <Input

                                        id="exampleCheck3"

                                        name="language"

                                        type="checkbox"

                                        className="language me-2"

                                        value='react'

                                        onChange={mainData}

                                        checked={obj.language.includes('react')}

                                    />

                                    <Label

                                        check

                                        for="exampleCheck3"

                                        className="px-2"

                                    >

                                        REACT

                                    </Label>

                                </div>

                            </div>

                        </Col>

                        <Col md={6}>

                            <Label for="example" className="fw-600 fs-5">

                                Gender

                            </Label>

                            <div className="d-flex">

                                <div>

                                    <Input

                                        id="exampleCheck3"

                                        name="gender"

                                        type="radio"

                                        className="gender me-2"

                                        value='male'

                                        onChange={mainData}

                                        checked={obj.gender === 'male'}

                                    />

                                    <Label

                                        check

                                        for="radio"

                                        className="px-2"

                                    >

                                        Male

                                    </Label>

                                </div>

                                <div>

                                    <Input

                                        id="exampleCheck3"

                                        name="gender"

                                        type="radio"

                                        className="gender me-2"

                                        value='female'

                                        onChange={mainData}

                                        checked={obj.gender === 'female'}

                                    />

                                    <Label

                                        check

                                        for="radio"

                                        className="px-2"

                                    >

                                        Female

                                    </Label>

                                </div>

                            </div>

                        </Col>

                        <Col md={6} className="imgheight">

                            <Label

                                check

                                for=""

                                className="py-2 fs-5"

                            >

                                Profile

                            </Label>

                            <div>

                                <input type="file" id="profile" name="profile" onChange={mainData} />

                            </div>

                        </Col>

                        <Col md={6}>

                            <div className="my-2">

                                {(obj.profile) && < img src={obj.profile} alt="img" className="object-fit-contain" width={200} height={200} />}

                            </div>

                        </Col>

                    </Row>

                    <div className="text-center">

                        <button className="my-2 btn btn-secondary">

                            Submit

                        </button>

                    </div>

                </Form>

            </Container>

            <Container>

                {console.log(array)}

                <MainTable mainArray={array} editfun={editfun} deletefun={deletefun} />

            </Container>

        </Fragment >

    )

}

export default MainForm

**Main table Component**

import React, { Fragment } from 'react'

import { Table } from 'reactstrap'

function MainTable(props) {

  return (

    <Fragment>

      <div className='my-5'>

        <h1 className='text-center py-3'>Empolyee Information</h1>

        <Table className='p-3'>

          <thead>

            <tr>

              <th>

                Sr No

              </th>

              <th>

                Profile

              </th>

              <th>

                Name

              </th>

              <th>

                Email

              </th>

              <th>

                Pin Code

              </th>

              <th>

                Phone No

              </th>

              <th>

                Gender

              </th>

              <th>

                Language

              </th>

              <th>

                Action

              </th>

            </tr>

          </thead>

          <tbody>

            {props.mainArray.map((x, i) => {

              return <tr key={i}>

                <td>{x.id}</td>

                <td>

                  <img src={x.profile} width={100} height={100} alt="img" />

                </td>

                <td>{x.username}</td>

                <td>{x.email}</td>

                <td>{x.pincode}</td>

                <td>{x.phoneno}</td>

                <td>{x.gender}</td>

                <td>{x.language.join(",")}</td>

                <td>

                  <button className="me-2" onClick={() => props.editfun(x.id)}>Edit</button>

                  <button onClick={() => props.deletefun(x.id)}>Delete</button>

                </td>

              </tr>

            })}

          </tbody>

        </Table>

      </div>

    </Fragment>

  )

}

export default MainTable

**App.js**

import 'bootstrap/dist/css/bootstrap.min.css';

import './App.css';

import MainForm from './components/MainForm';

import MainTable from './components/MainTable';

function App() {

  return (

    <div>

      <MainForm />

    </div>

  );

}

export default App;

**Child to data pass**

**Parent Component**

import React, { useState } from 'react'

import ChildComp from './ChildComp'

function ParentComp() {

    const [number, setnumber] = useState(1234)

    /\* to take data from child to parent , we use function which we define in parent component and pass through props in child component and call in child component with argument which we want in parent component and then use that from function parameter  \*/

    const dataFun = (a) => {

        setnumber(a)

    }

    return (

        <div>

            <ChildComp data={dataFun} />

            <h1>{number}</h1>

        </div>

    )

}

export default ParentComp

**Child Component**

import React from 'react'

function ChildComp(props) {

    return (

        <div>

            <h1>

                {props.data(2000)}

            </h1>

        </div>

    )

}

export default ChildComp

**App.js**

import './App.css';

import ParentComp from './components/ParentComp';

function App() {

  return (

    <div className="App">

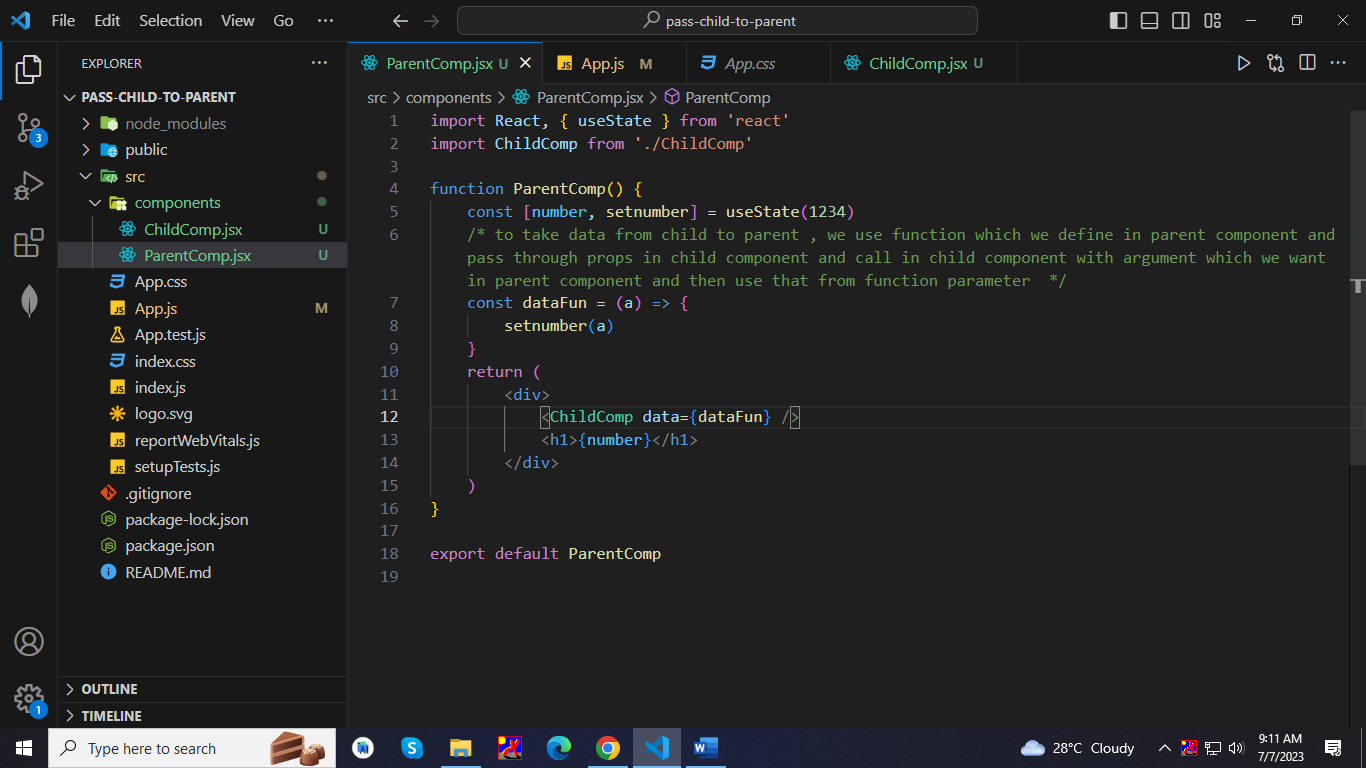
      <ParentComp />

    </div>

  );

}

export default App;



**Useref**

import React, { useEffect, useRef } from 'react'

function Useref() {

    const reference = useRef()

    /\* useref is hook

    useref is use to select element from html

    then we can do any manipulation we want.

    stateName.current give element\*/

    useEffect(() => {

        console.log(reference.current)

        reference.current.style.backgroundColor = 'red'

    }, [])

    return (

        <div>

            <h1 ref={reference}>Hello world</h1>

        </div>

    )

}

export default Useref

**Output**

