

Advance JS crud

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>
        <!-- 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>
                    </div>
                </div>
            </div>
        </form>
    </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 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"
for="flexCheckDefault">Female</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input radio"
type="radio" value="trans"
for="flexCheckDefault">Trans</label>
</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 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>
<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.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;
}


```

```
#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 successfully 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  
format 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 Anynome 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';
```

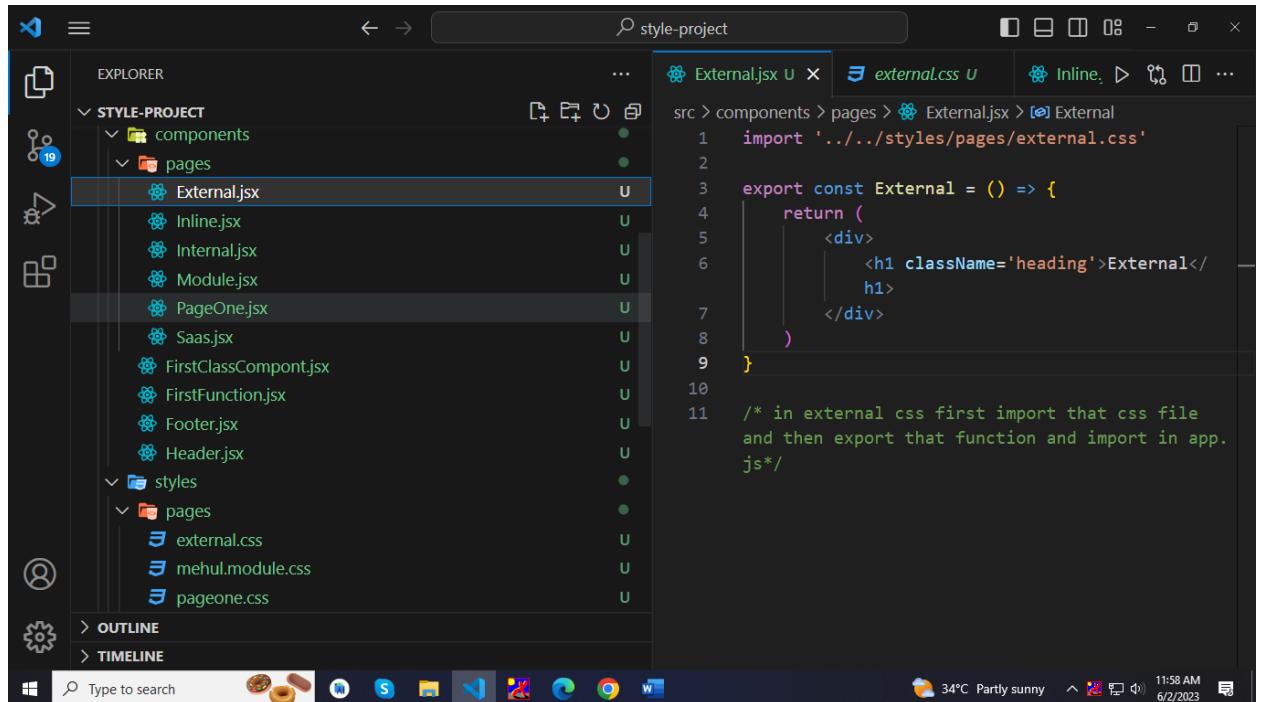
2=>Jo export default karyu hoy to

```
import FirstClassComponent from './components/FirstClassCompont';  
ane aapne FirstClassComponent ni jayae koi biju name  
pan rakhi shakiye.
```

2)Jo file Import karvi hoy to

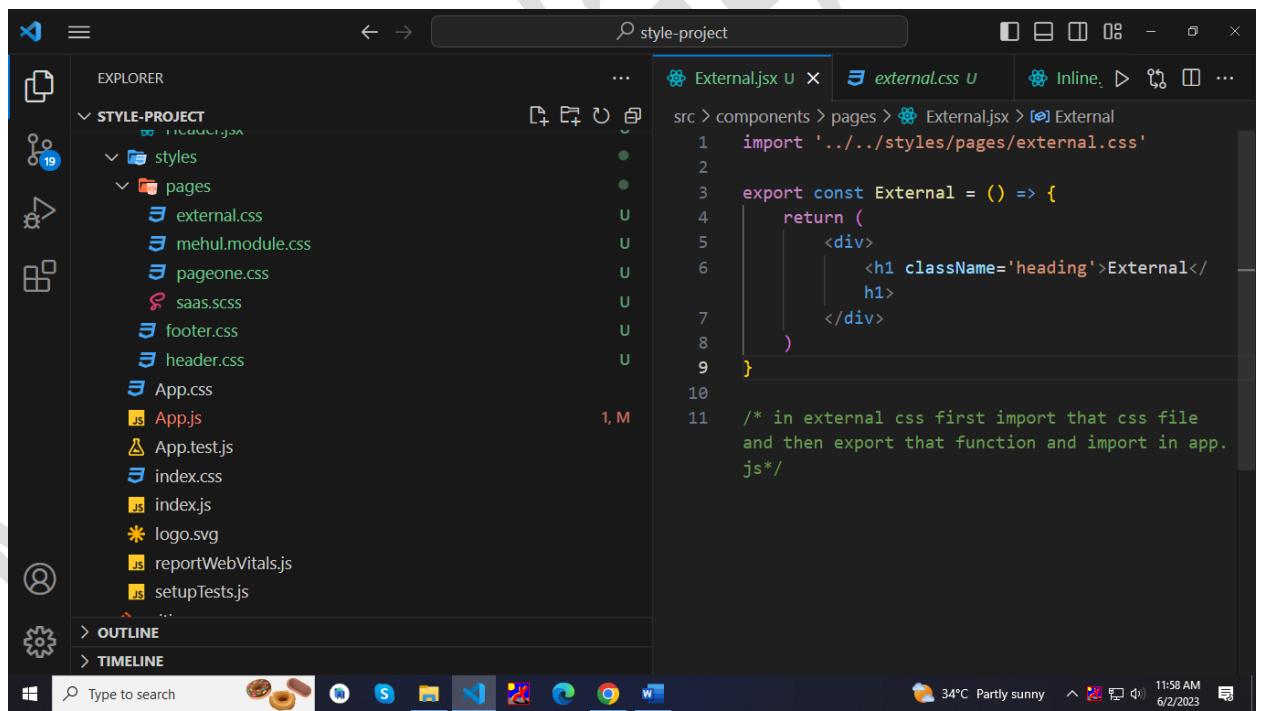
```
import './App.css';
```

Folder Structure



The screenshot shows the VS Code interface with the title bar "style-project". The Explorer sidebar on the left displays a project structure under "STYLE-PROJECT". The "components" folder contains "External.jsx", "Inline.jsx", "Internal.jsx", "Module.jsx", "PageOne.jsx", "Saas.jsx", "FirstClassComponet.jsx", "FirstFunction.jsx", "Footer.jsx", and "Header.jsx". The "pages" folder contains "External.jsx", "Inline.jsx", "Internal.jsx", "Module.jsx", "PageOne.jsx", "Saas.jsx", "FirstClassComponet.jsx", "FirstFunction.jsx", "Footer.jsx", and "Header.jsx". The "styles" folder contains "pages" which includes "external.css", "mehul.module.css", and "pageone.css". The "External.jsx" file in the "pages" folder is currently selected and open in the main editor area. The code in the editor is:

```
src > components > pages > External.jsx > [External]
1 import '../../../../../styles/pages/external.css'
2
3 export const External = () => {
4   return (
5     <div>
6       <h1 className='heading'>External</h1>
7     </div>
8   )
9 }
10
11 /* in external css first import that css file
12  and then export that function and import in app.
13  js*/
```



The screenshot shows the VS Code interface with the title bar "style-project". The Explorer sidebar on the left displays a project structure under "STYLE-PROJECT". The "components" folder contains "header.jsx". The "styles" folder contains "pages" which includes "external.css", "mehul.module.css", "pageone.css", "saas.scss", "footer.css", and "header.css". The "App.css", "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", and "setupTests.js" files are also listed. The "External.jsx" file in the "components" folder is currently selected and open in the main editor area. The code in the editor is identical to the one in the first screenshot.

Inline Style

A screenshot of a code editor window titled "style-project". The current file is "Inline.jsx". The code is as follows:

```
src > components > pages > Inline.jsx > ...
1  export const Inline =() =>{
2    return(
3      <div>
4        <h1 style={{color:'blue',backgroundColor:'pink'}}
5          >Inline Css</h1>
6        </div>
7    )
8  /* inline css ma style lakhi object ni under object banavi pachi
   style aapvi pade ane property na name ma jo Hyphen aave to a
   property ne hyphen lagadya sivaay camel case ma lakhvi*/
```

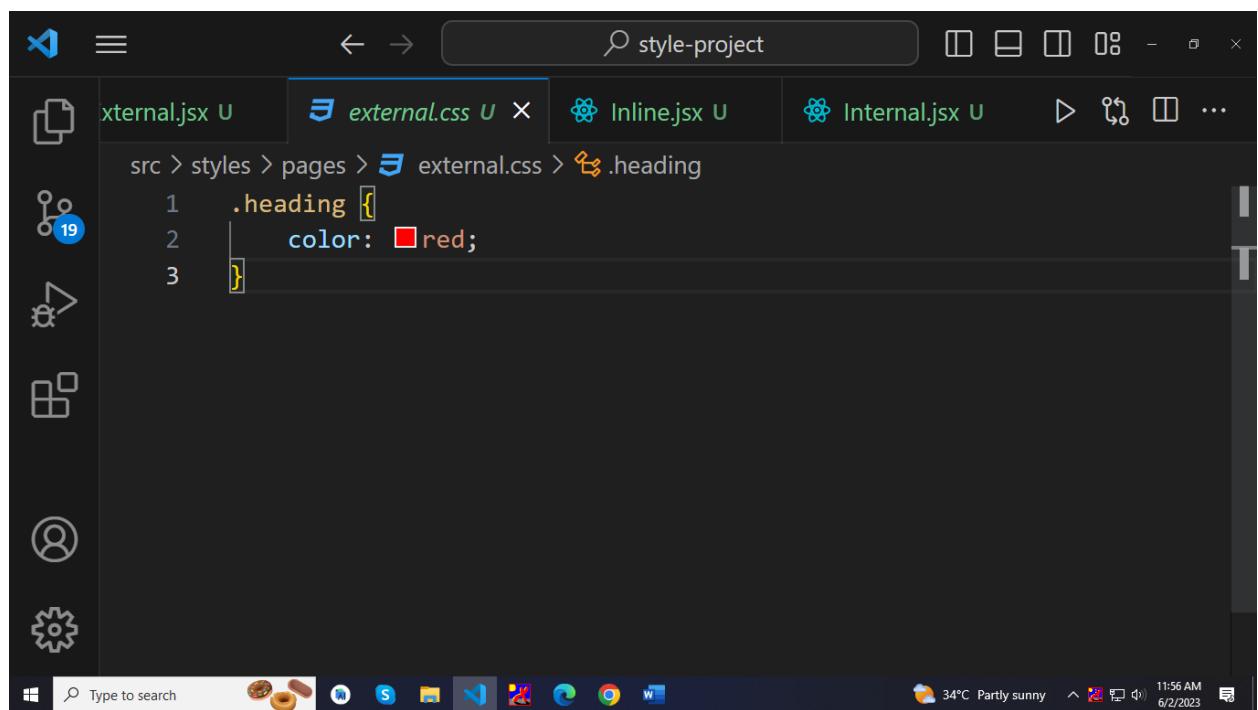
The status bar at the bottom shows the date and time as 6/2/2023, 11:55 AM.

External Style

A screenshot of a code editor window titled "style-project". The current file is "External.jsx". The code is as follows:

```
src > components > pages > External.jsx > [External]
1  import '../styles/pages/external.css'
2
3  export const External = () => [
4    return (
5      <div>
6        <h1 className='heading'>External</h1>
7      </div>
8    )
9  ]
10
11 /* in external css first import that css file and then export that
   function and import in app.js*/
```

The status bar at the bottom shows the date and time as 6/2/2023, 11:56 AM.



```
src > styles > pages > external.css > .heading
1 .heading {
2   color: red;
3 }
```

Module CSS

A screenshot of the Visual Studio Code interface. The title bar says "style-project". The left sidebar has icons for file, folder, search, and settings. The main editor shows the "Module.js" file:

```
src > components > pages > Module.js > ...
1 import styles from '../.../styles/pages/mehul.module.css';
2 export const Module = ()=>{
3     return (
4         <div>
5             <h1 className={styles.main}>Module Style</h1>
6         </div>
7     )
8 }
/* Module thi css add kariye tyare css file bane anu formate =>
anyname.module.css rakhvu compulsory che
anu extension compulsory module.css aavse */
/* jya use kariye a style tya je name thi import karyu hoy a name
thi classname ma as a object use karvu */
```

The status bar at the bottom shows the date and time.

A screenshot of the Visual Studio Code interface. The title bar says "style-project...". The left sidebar has icons for user and settings. The main editor shows the "mehul.module.css" file:

```
src > styles > pages > mehul.module.css > .main
1 .main{
2     color: cadetblue;
3 }
```

The status bar at the bottom shows the date and time.

Internal css

A screenshot of a code editor window titled "style-project". The current file is "Internal.js". The code defines a style object named "myStyle" containing two nested objects: "mainHeading" and "subHeading". The "mainHeading" object has a color of "orange". The "subHeading" object has a border of "1px solid black", a color of "indianred", and a font size of "70px". An export statement creates a function "Internal" that returns a component with two headings. The "style-project" search bar is visible at the top.

```
src > components > pages > Internal.jsx > ...
1  /* Internal css ma first ek object banavavo pade ama nested object banavi je style aapvi hoy a lakhvani
   ane use karo tyare style lakhi ama curaly bracket ma main object .ane je property lagavani hoy a */
2
3  const myStyle = {
4      mainHeading:{ 
5          color:'orange'
6      },
7      subHeading:{ 
8          border:"1px solid black",
9          color:"indianred",
10         fontSize:"70px"
11     }
12 }
13 export const Internal = () => {
14     return (
15         <div>
16             <h1 style={myStyle.mainHeading}>Internal Css</h1>
17             <h2 style={myStyle.subHeading}>Internal Sub heading</h2>
18         </div>
19     )
20 }
21
```

Saas CSS

A screenshot of a code editor window titled "style-project". The current file is "saas.scss". It contains SCSS code defining a variable "\$hello" with a value of "indianred" and a class ".head" with a color of "\$hello". A note at the bottom indicates that to run the project, "saas" must be installed via npm. The "style-project" search bar is visible at the top.

```
src > styles > pages > saas.scss > ...
1  $hello : indianred;
2  .head{
3      color:$hello;
4  }
5 // To Run Saas first install saas from terminal by
6 // this command
7 // npm i saas or npm inastall saas|
```

```
src > components > pages > Saas.jsx > ...
1  /* when use saas first install saas by command=> npm i saas */
2  import '../../../../../styles/pages/saas.scss'
3  export const Saas = ()=>{
4      return(
5          <div>
6              <h1 className="head">Saas Style</h1>
7          </div>
8      )
9 }
```

App

Js file

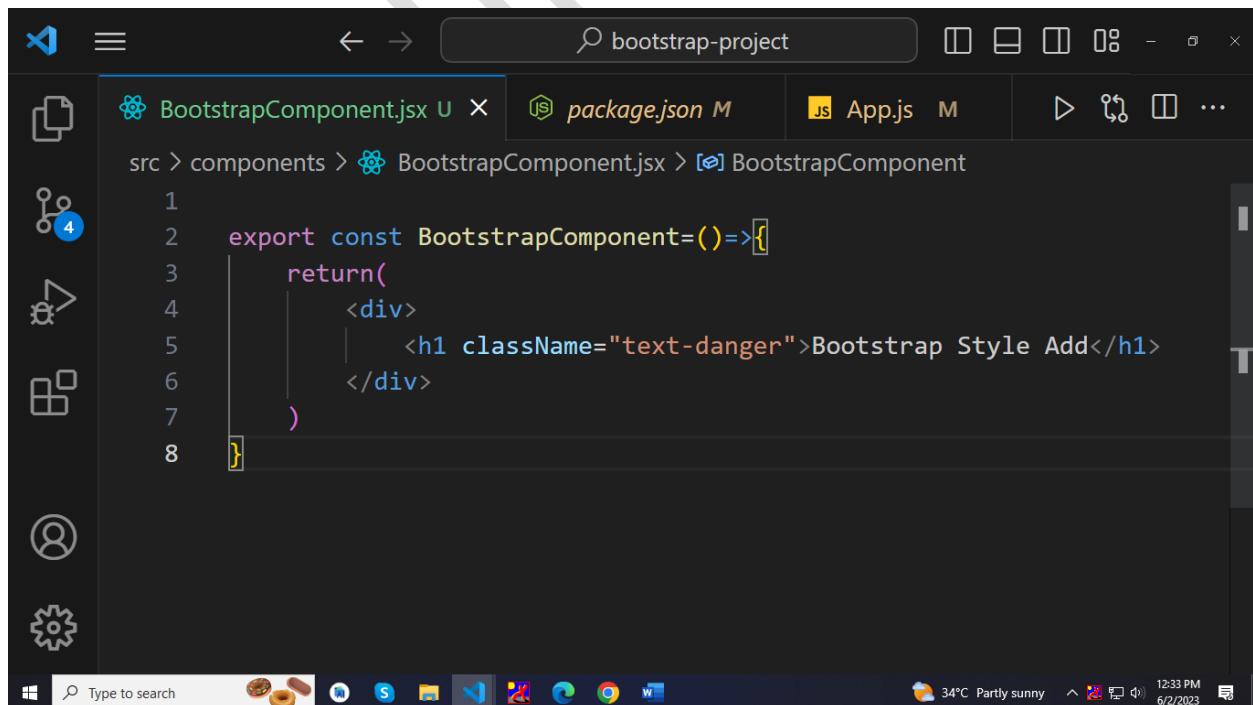
```
import './App.css';
import FirstClassComponent from './components/FirstClassCompont';

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;
```

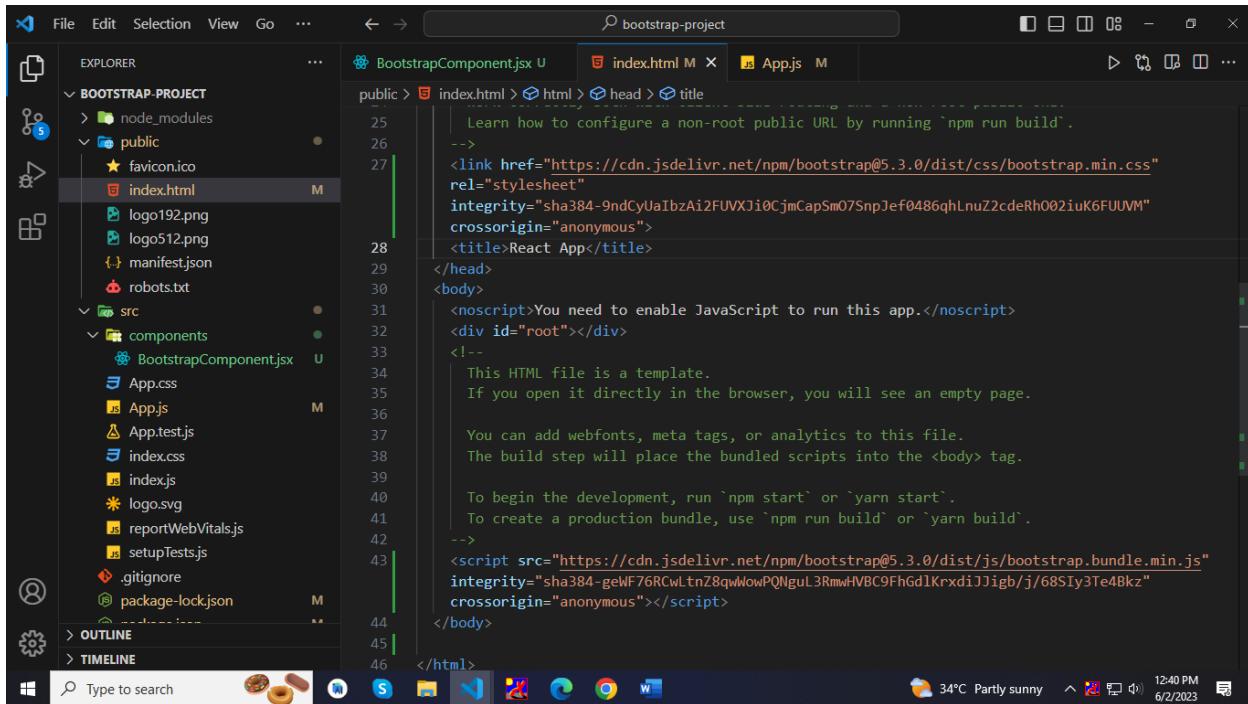
Bootstrap



The screenshot shows a code editor interface with a dark theme. The top bar includes icons for file operations, a search bar containing "bootstrap-project", and window control buttons. The left sidebar features a tree view with four items under "src > components". The main editor area displays the following code:

```
1  export const BootstrapComponent=()=>{
2      return(
3          <div>
4              <h1 className="text-danger">Bootstrap Style Add</h1>
5          </div>
6      )
7  }
```

The code defines a functional component named `BootstrapComponent`. It returns a single `<div>` element containing a `<h1>` element with the class `text-danger` and the text "Bootstrap Style Add". The code editor's status bar at the bottom shows system information like weather (34°C Partly sunny), time (12:33 PM), and date (6/2/2023).



```
public > index.html > html > head > title
  Learn how to configure a non-root public URL by running `npm run build`.
  -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdRh002iuK6FUUV"
crossorigin="anonymous">
<title>React App</title>
</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div>
<!--
  This HTML file is a template.
  If you open it directly in the browser, you will see an empty page.

  You can add webfonts, meta tags, or analytics to this file.
  The build step will place the bundled scripts into the <body> tag.

  To begin the development, run `npm start` or `yarn start`.
  To create a production bundle, use `npm run build` or `yarn build`.
-->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwlnZ8qvwPQNgUL3RmwHVBC0fhGdIKrxdiJJgib/j68SiY3Te4Bkz" crossorigin="anonymous"></script>
</body>
</html>
```

In index.html file bootstrap css file 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 parameter like this using anonymous
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;
```

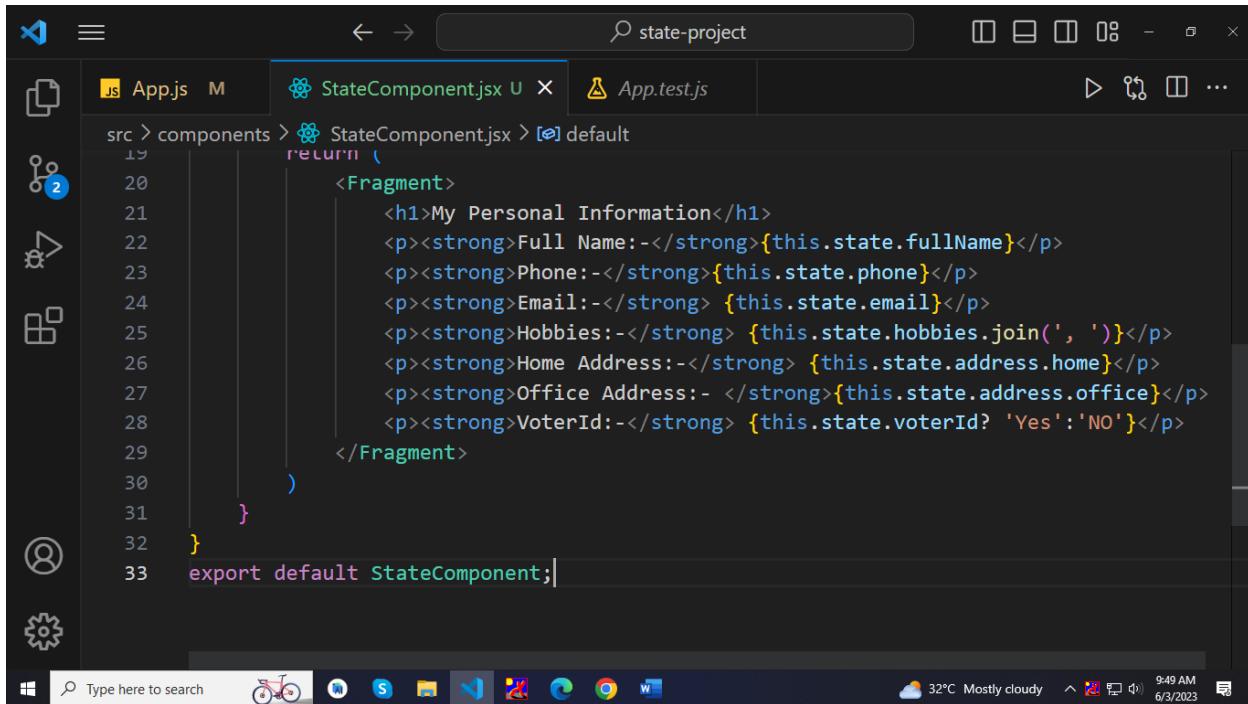
A screenshot of a dark-themed code editor window titled "state-project". The current file is "StateComponent.jsx". The code defines a class component "StateComponent" that extends "Component". It has a constructor that initializes "this.state" with personal information: fullName, phone, email, hobbies, address, and voterId. The "address" object contains "home" and "office" properties.

```
src > components > StateComponent.jsx > [o] default
1 import React, { Component, Fragment } from 'react'
2
3 class StateComponent extends Component {
4     constructor() {
5         super()
6         this.state = {
7             fullName: 'Mehul Balvantbhai Waghela',
8             phone: 6355173787,
9             email: 'mehulwaghela63@gmail.com',
10            hobbies: ['Dancing', 'Travelling', 'Coding'],
11            address: {
12                home: '71,Deepnagar society,kim',
13                office: 'Oscar Career Point'
14            },
15            voterId: true
16        }
17    }

```

A screenshot of the same dark-themed code editor window, showing the addition of the "render()" method to the "StateComponent" class. The "render()" method returns a "Fragment" containing four "p" elements, each with a "strong" tag and dynamic content from "this.state".

```
src > components > StateComponent.jsx > [o] default
9             email: 'mehulwaghela63@gmail.com',
10            hobbies: ['Dancing', 'Travelling', 'Coding'],
11            address: {
12                home: '71,Deepnagar society,kim',
13                office: 'Oscar Career Point'
14            },
15            voterId: true
16        }
17    }
18    render() {
19        return (
20            <Fragment>
21                <h1>My Personal Information</h1>
22                <p><strong>Full Name:-</strong>{this.state.fullName}</p>
23                <p><strong>Phone:-</strong>{this.state.phone}</p>
24                <p><strong>Email:-</strong> {this.state.email}</p>
25                <p><strong>Hobbies:-</strong> {this.state.hobbies.join(', ')})</p>
```



```
src > components > StateComponent.jsx > [o] default
19   return (
20     <Fragment>
21       <h1>My Personal Information</h1>
22       <p><strong>Full Name:</strong> {this.state.fullName}</p>
23       <p><strong>Phone:</strong> {this.state.phone}</p>
24       <p><strong>Email:</strong> {this.state.email}</p>
25       <p><strong>Hobbies:</strong> {this.state.hobbies.join(', ')</p>
26       <p><strong>Home Address:</strong> {this.state.address.home}</p>
27       <p><strong>Office Address:</strong> {this.state.address.office}</p>
28       <p><strong>VoterId:</strong> {this.state.voterId? 'Yes': 'NO'</p>
29     </Fragment>
30   )
31 }
32 }
33 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 receive</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} />
```

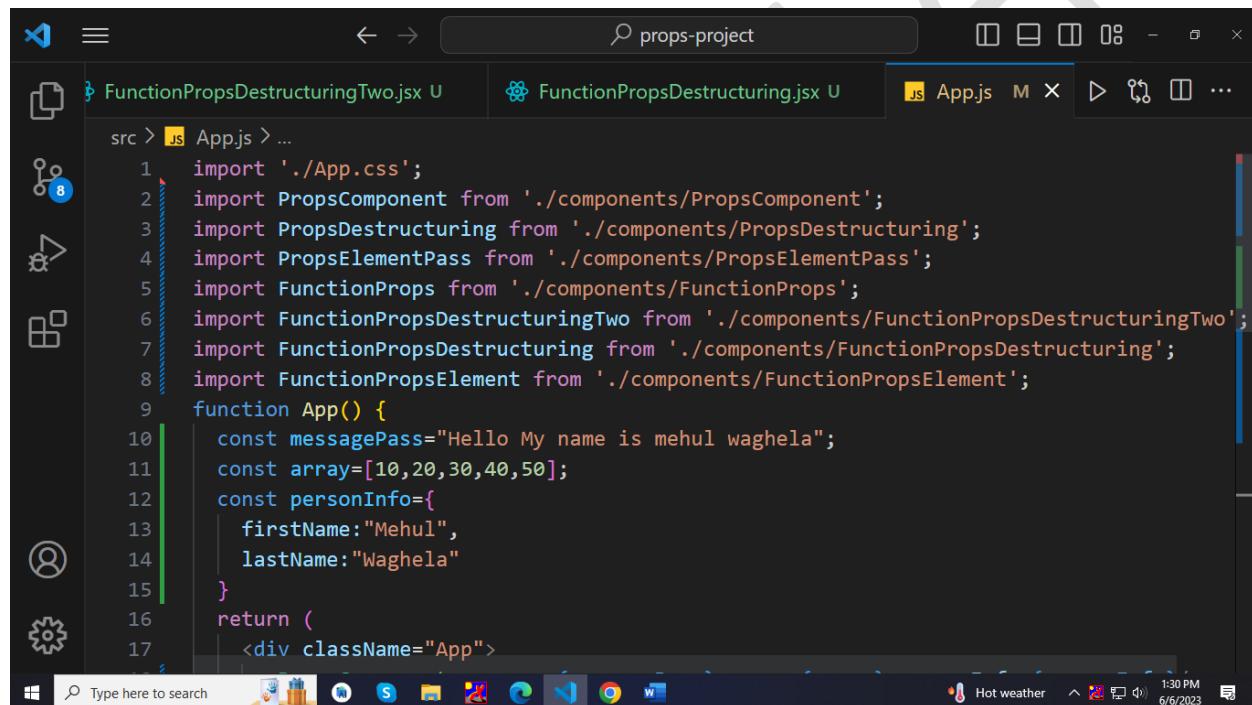
Function props element pass

```
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



The screenshot shows a Windows desktop environment with a code editor window open. The title bar of the window reads 'props-project'. The code editor displays the file 'App.js' with the following content:

```
src > js App.js > ...  
1 import './App.css';  
2 import PropsComponent from './components/PropsComponent';  
3 import PropsDestructuring from './components/PropsDestructuring';  
4 import PropsElementPass from './components/PropsElementPass';  
5 import FunctionProps from './components/FunctionProps';  
6 import FunctionPropsDestructuringTwo from './components/FunctionPropsDestructuringTwo';  
7 import FunctionPropsDestructuring from './components/FunctionPropsDestructuring';  
8 import FunctionPropsElement from './components/FunctionPropsElement';  
9 function App() {  
10     const messagePass="Hello My name is mehul waghela";  
11     const array=[10,20,30,40,50];  
12     const personInfo={  
13         firstName:"Mehul",  
14         lastName:"Waghela"  
15     }  
16     return (  
17         <div className="App">
```

The code editor interface includes a sidebar with icons for file operations like Open, Save, and Delete, and a status bar at the bottom showing the date and time.



```
src > js App.js > [o] default
14 |   lastName:"Waghela"
15 |
16 |   return (
17 |     <div className="App">
18 |       <PropsComponent message={messagePass} array={array} personInfo={personInfo}/>
19 |       <PropsDestructuring message={messagePass} array={array} personInfo={personInfo}/>
20 |       <PropsElementPass>
21 |         <p>Hello This Is paragraph</p>
22 |       </PropsElementPass>
23 |       <FunctionProps message={messagePass} array={array} personInfo={personInfo} />
24 |       <FunctionPropsDestructuring message={messagePass} array={array} personInfo={personInfo} />
25 |       <FunctionPropsDestructuringTwo message={messagePass} array={array} personInfo={personInfo} />
26 |       <FunctionPropsElement>
27 |         <p>Function based Element pass props</p>
28 |       </FunctionPropsElement>
29 |     </div>
30   );
31 }
32 export default App;
33
```



EXPLORER

- PROPS-PROJECT
 - > node_modules
 - > public
 - < src
 - < components
 - FunctionProps.jsx
 - FunctionPropsDestructuring.jsx
 - FunctionPropsDestructuringTwo.jsx
 - FunctionPropsElement.jsx
 - PropsComponent.jsx
 - PropsDestructuring.jsx
 - PropsElementPass.jsx
 - App.css
 - js App.js
 - App.test.js
 - index.css
 - js index.js
 - * logo.svg

```
src > components > PropsComponent.jsx > PropsComponent
1 import React, {Component,Fragment} from 'react'
2
3 class PropsComponent extends Component{
4
5   render(){
6     console.log(this.props);
7     return(
8       <Fragment>
9         <h1>Props Component Simple recieve</h1>
10        <p>{this.props.message}</p>
11        <p>{this.props.array}</p>
12        <p>{this.props.personInfo.firstName}</p>
13      </Fragment>
14    )
15  }
16}
17 export default PropsComponent
```



A screenshot of a Windows desktop showing a Microsoft Visual Studio Code (VS Code) window. The window title is "props-project". The Explorer sidebar on the left shows a project structure with files like "PropsComponent.jsx", "PropsDestructuring.jsx", and "PropsElement.jsx" under the "src/components" folder. The "PropsDestructuring.jsx" file is open in the main editor area, displaying the following code:

```
1 import React, {Component,Fragment} from 'react'
2
3 class PropsDestructuring extends Component{
4   render(){
5     const {firstName,lastName} =this.props.personInfo;
6     return(
7       <Fragment>
8         <h1>Props Component destructuring recieve</h1>
9         <p>{firstName}</p>
10        <p>{lastName}</p>
11      </Fragment>
12    )
13  }
14
15 export default PropsDestructuring
```

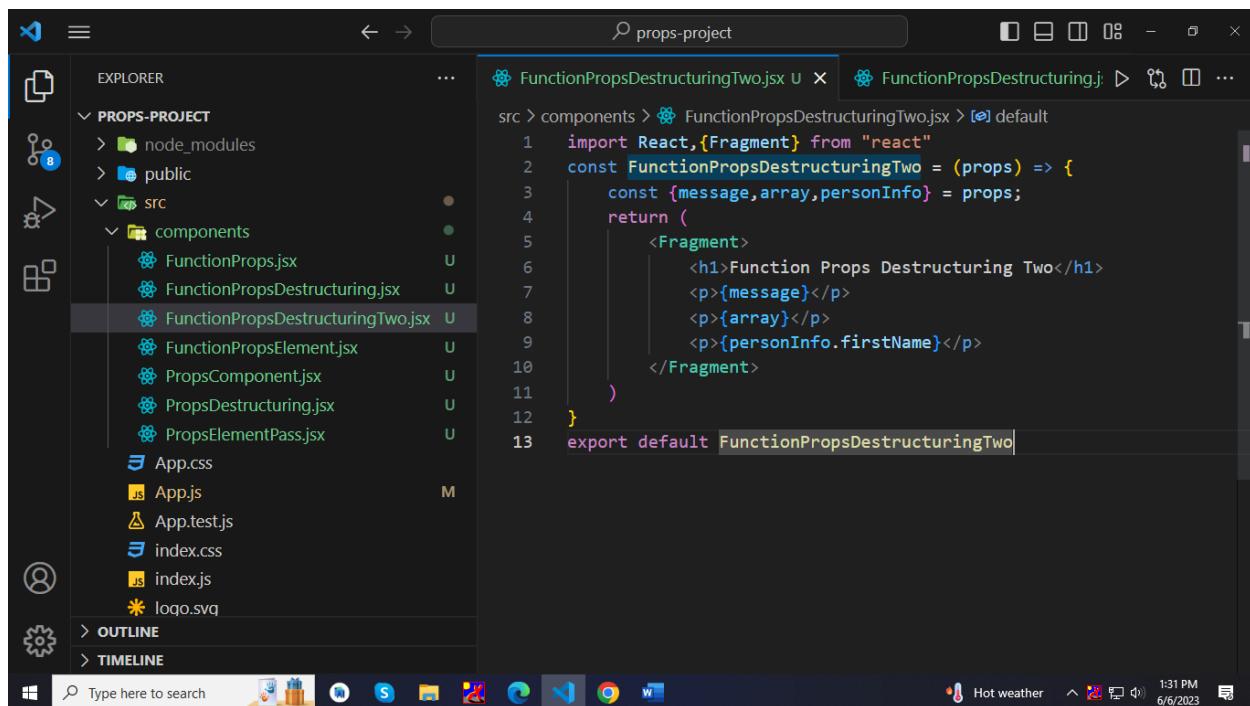
The status bar at the bottom shows the date and time as "6/6/2023 1:32 PM".

The screenshot shows the VS Code interface with the title bar "props-project". The Explorer sidebar on the left shows a project structure with files like App.css, App.js, App.test.js, index.css, index.js, and logo.svg. The main editor area displays the code for `FunctionProps.jsx`:

```
src > components > FunctionProps.jsx > [default]
1 import React,{Fragment} from "react"
2 const FunctionProps = (props) => {
3     return (
4         <Fragment>
5             <h1>Function Props Pass simple</h1>
6             <p>{props.message}</p>
7             <p>{props.array}</p>
8             <p>{props.userInfo.firstName}</p>
9         </Fragment>
10    )
11 }
12 export default FunctionProps
```

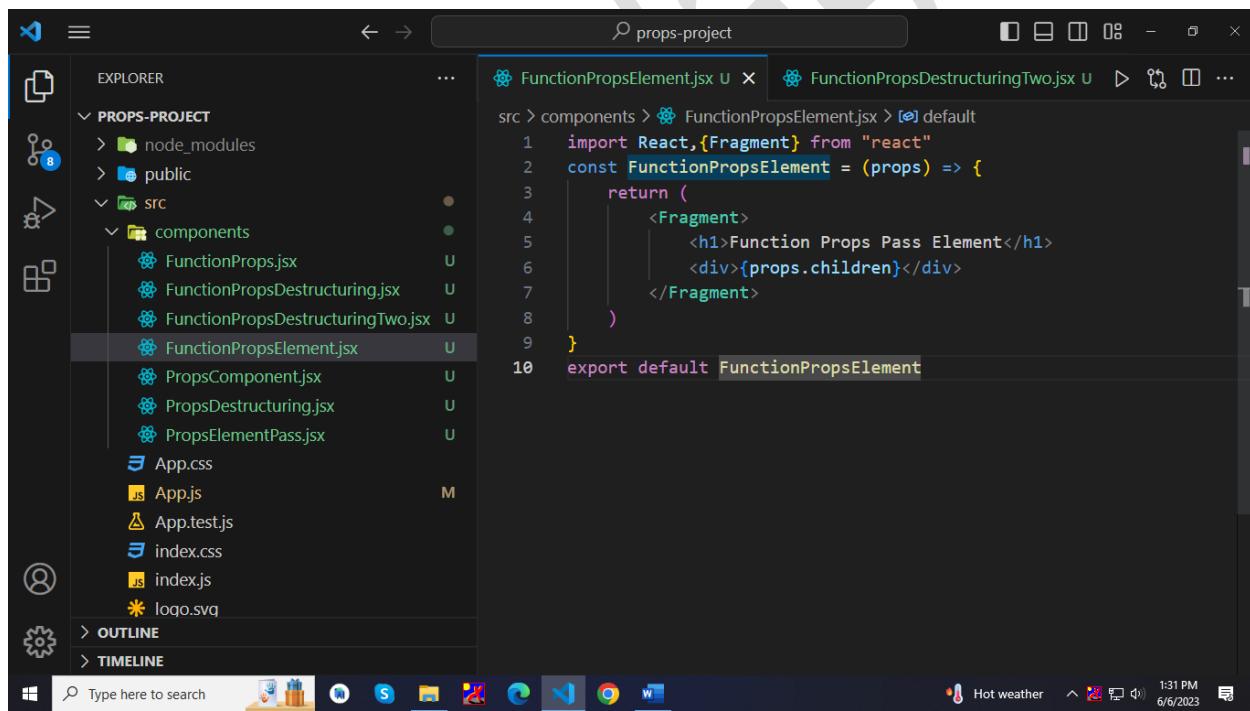
The screenshot shows the VS Code interface with the title bar "props-project". The Explorer sidebar on the left shows a project structure with files like App.css, App.js, App.test.js, index.css, index.js, and logo.svg. The main editor area displays the code for `FunctionPropsDestructuring.jsx`:

```
src > components > FunctionPropsDestructuring.jsx > [default]
1 import React,{Fragment} from "react"
2 const FunctionPropsDestructuring = ({message}) => {
3     return (
4         <Fragment>
5             <h1>Function Props parameter destructuring</h1>
6             <p>{message}</p>
7         </Fragment>
8     )
9 }
10 export default FunctionPropsDestructuring
```



props-project

```
src > components > FunctionPropsDestructuringTwo.jsx > default
1 import React,{Fragment} from "react"
2 const FunctionPropsDestructuringTwo = (props) => {
3   const {message,array,personInfo} = props;
4   return (
5     <Fragment>
6       <h1>Function Props Destructuring Two</h1>
7       <p>{message}</p>
8       <p>{array}</p>
9       <p>{personInfo.firstName}</p>
10    )
11  }
12 }
13 export default FunctionPropsDestructuringTwo
```



props-project

```
src > components > FunctionPropsElement.jsx > default
1 import React,{Fragment} from "react"
2 const FunctionPropsElement = (props) => {
3   return (
4     <Fragment>
5       <h1>Function Props Pass Element</h1>
6       <div>{props.children}</div>
7     </Fragment>
8   )
9 }
10 export default FunctionPropsElement
```

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>  
                                </Card>  
                            </Col>  
                        )  
                    })  
                </Row>  
            </Container>  
    )  
}
```

```

        <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 COnponentDidMount 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

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure under "PURE-COMPONENT". It includes a "node_modules" folder, a "public" folder, and a "src" folder containing "components", "App.css", "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", "setupTests.js", ".gitignore", "package-lock.json", "package.json", and "README.md".
- CODE EDITOR:** The "ChildComponent.jsx" file is open. The code is as follows:

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

export class ChildComponent extends Component {

  render() {
    console.log("Child Component");
    return (
      <div>{this.props.message}</div>
    )
  }
}

export default ChildComponent
```

- TERMINAL:** Shows the output of the build process:

```
You can now view pure-component in the browser.

Local:          http://localhost:3000
On Your Network: http://192.168.1.51:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```
- STATUS BAR:** Shows the date (6/13/2023), time (11:44 AM), and weather (33°C Mostly cloudy).

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
```

Mehul

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((element) => element.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>
    </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) ?? ''} />
      </div>
    </Col>
  </Row>

```

```

                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.h
obby.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>
</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 */
    componentWillMount() {
        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 fluid>
        </Fragment>
    )
}

```

```

        </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>
                            }
                </Row>
            </ModalBody>
        </Modal>
    
```

```

        <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.g
ender === '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.g
ender === '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.g
ender === 'trans'} />
              <Label for="radioThree"
                className='ms-2'>Trans</Label>
            </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')}}

        </div>
        <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.h
obby.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>

```

```
        )
      })
    </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 available 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 lakhya 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 dependency
*/
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, setMessage] = useState({
    userMessage: 'Guest user',
    loginState: 'Please Login'
  })
  const changeLogin = () => {
    setLogin(!login)
    if (login !== true) {
      setMessage({
        userMessage: 'Welcome Mehul Waghela',
        loginState: 'You are Online'
      })
    }
    else {
      setMessage({
        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>  
            <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;
```

ErrorBoundary 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
        </div>
      </div>
    </div>
  )
}

export default Style
```

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
```

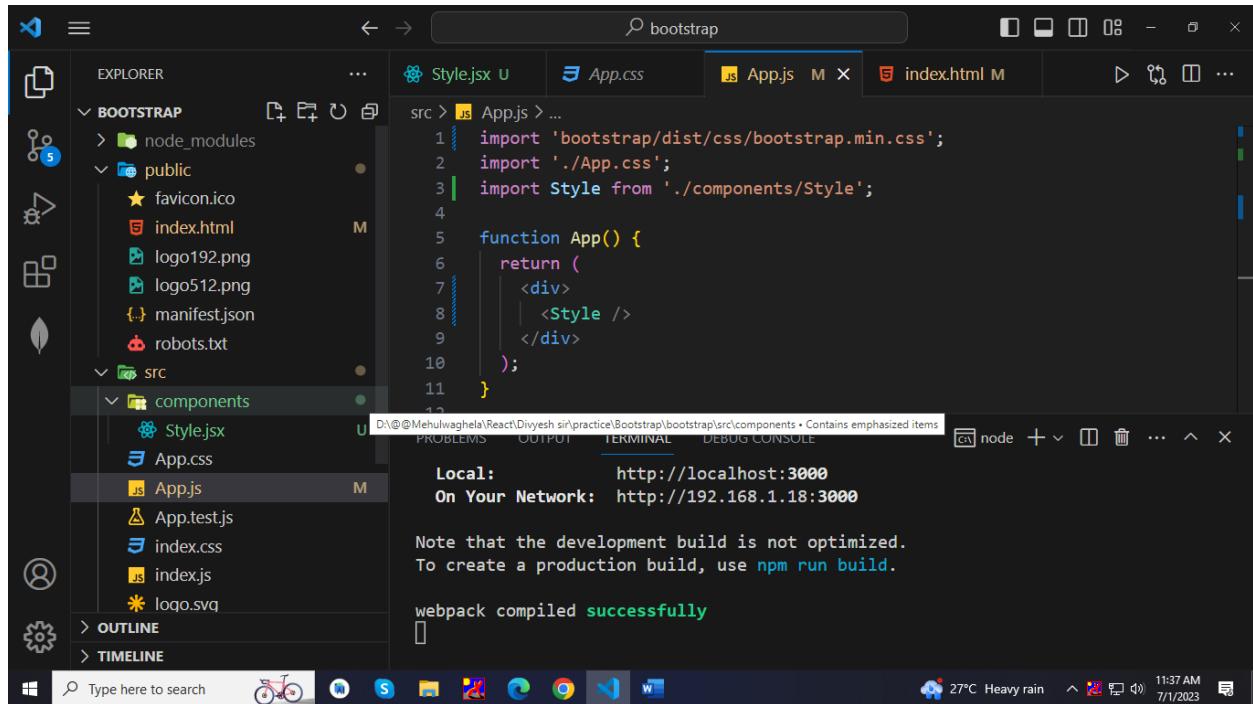
App.js

```
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

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER View:** Shows the project structure under "REACTBOOTSTRAP". The "src" folder contains "Components", "Style.jsx", "App.css", "App.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", and "setupTests.js".
- CODE EDITOR View:** The "App.js" file is open. The code is as follows:

```
src > js App.js > ...
1 import 'bootstrap/dist/css/bootstrap.min.css';
2 import './App.css';
3 import Style from './Components/Style';

4 function App() {
5   return (
6     <div className="App">
7       <Style/>
8     </div>
9   );
10 }
11 
12 export default App;
13 
14 export default App;
```

The code editor shows syntax highlighting for JavaScript and CSS. A tooltip is visible over the word "App" in the final line of the code.

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} />
      </div>
    </div>
  )
}
```

```
        <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 color="warning"
                  inverse
                  style={{
                      width: '18rem'
                  }}>
                <img
                    alt="Sample"
                    src={props.image}
                    style={{ objectFit: 'cover', width: "100%", height: '250px' }}>
            </Card>
            <CardBody>
                <CardTitle tag="h5">
                    {props.title}
                </CardTitle>
                <CardSubtitle
                    className="mb-2 text-muted"
                    tag="h6">
                </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 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

The screenshot shows a Windows desktop environment with two instances of Visual Studio Code (VS Code) running side-by-side. Both instances have dark themes applied.

Top Window (VS Code):

- Explorer:** Shows a project structure under the 'STYLE' folder:
 - node_modules
 - public
 - src
 - components (contains Css.jsx)
 - style
 - abc.scss
 - button.module.css
 - external.css
 - App.css
- Code Editor:** Displays the contents of 'App.js'. The code imports styles from 'App.css' and defines a functional component 'App' that returns a single

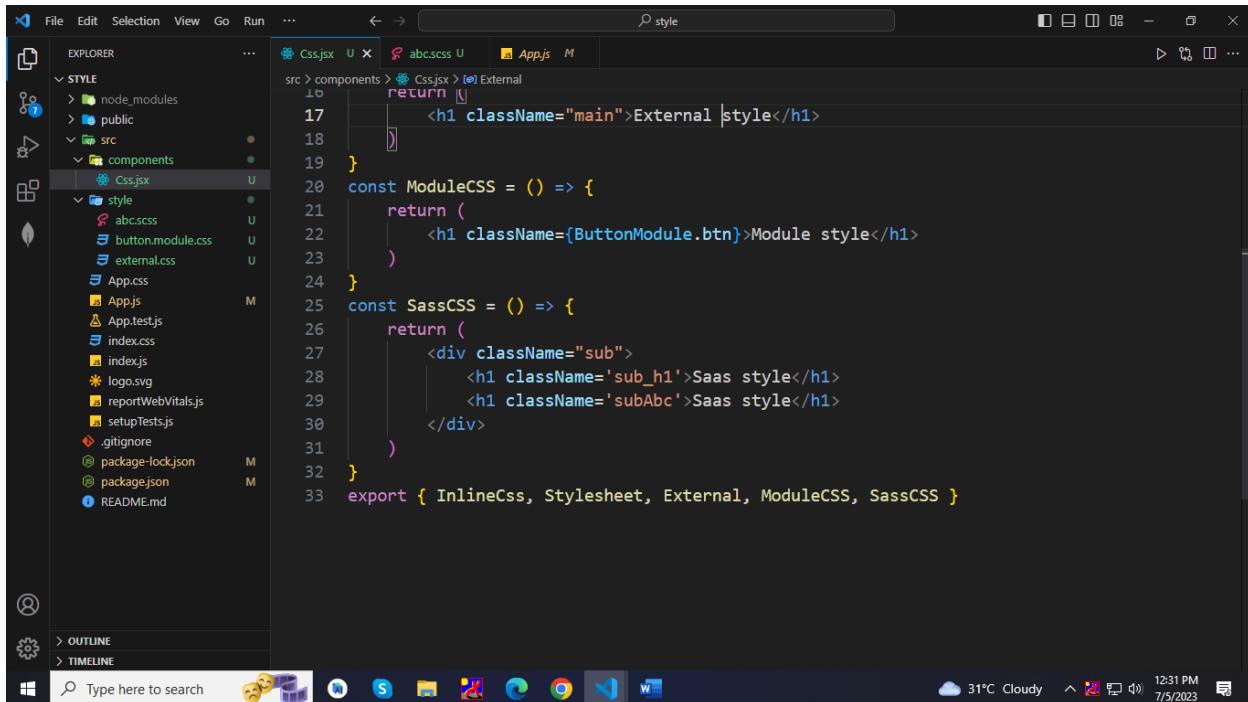
element with a class name of 'App'. Inside this div, there are four nested components: `<InlineCss />`, `<Stylesheet />`, `<External />`, and `<ModuleCSS />`. The final line exports the 'App' function.

```
src > App.js > ...
1 import './App.css';
2 import { InlineCss, Stylesheet, External, ModuleCSS, SassCSS } from './components/Css';
3
4 function App() {
5   return (
6     <div className="App">
7       <InlineCss />
8       <Stylesheet />
9       <External />
10      <ModuleCSS />
11    </div>
12  );
13}
14
15 export default App;
16
17
```

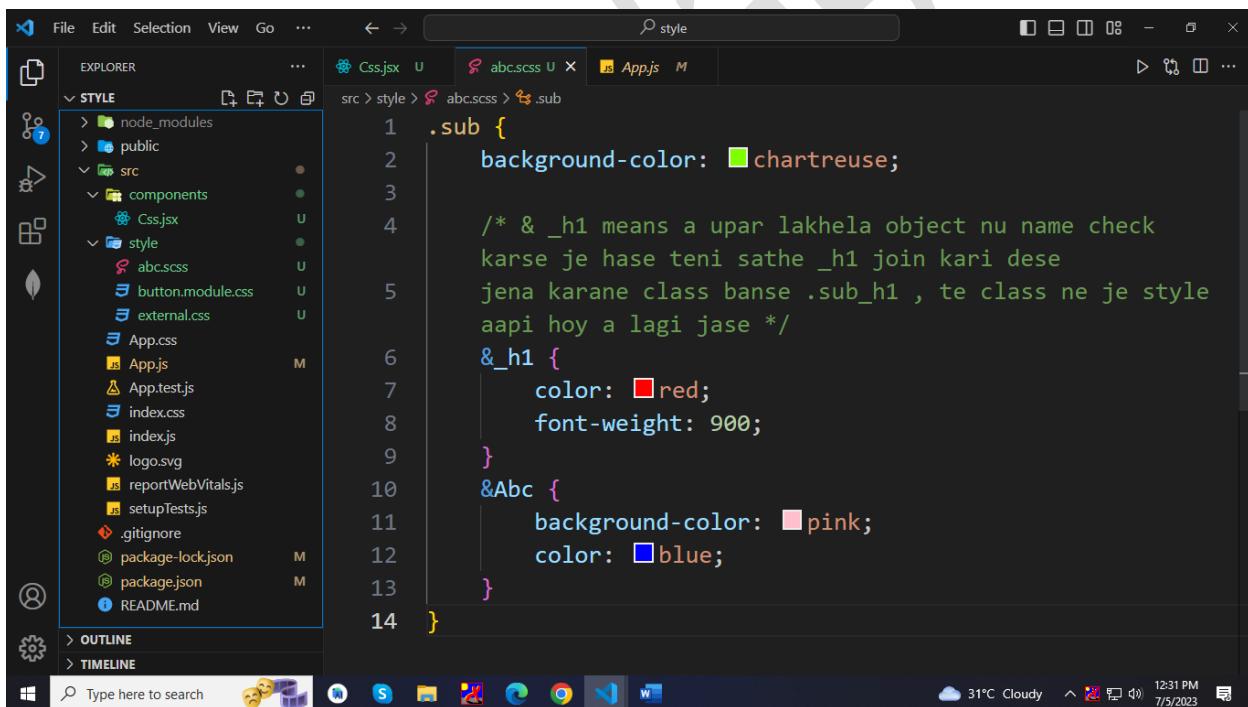
Bottom Window (VS Code):

 - Explorer:** Shows a similar project structure, but the 'components' folder now contains 'Css.jsx' instead of 'Css.js'.
 - Code Editor:** Displays the contents of 'Css.jsx'. This file imports 'external.css' and 'abc.scss' from the 'style' folder. It then defines several functions: `InlineCss`, `Stylesheet`, `External`, `ModuleCSS`, and `SassCSS`. Each function returns a different type of styled component: `Inline style` (using inline styles), `stylesheet or inline style` (using a CSS object in the style attribute), `External style` (using a class name), and `Module style` (using a class name derived from a module).

```
src > components > Css.jsx > External
1 import '../style/external.css'
2 import '../style/abc.scss'
3 import ButtonModule from '../style/button.module.css'
4 const InlineCss = () => {
5   return (
6     <h1 style={{ backgroundColor: "yellow" }}>Inline style</h1>
7   )
8 }
9 const Stylesheet = () => {
10   let abc = { backgroundColor: "red" }
11   return (
12     <h1 style={abc}>stylesheet or inline style</h1>
13   )
14 }
15 const External = () => {
16   return [
17     <h1 className="main">External style</h1>
18   ]
19 }
20 const ModuleCSS = () => {
21   return (
22     <h1 className={ButtonModule.btn}>Module style</h1>
23   )
24 }
```



```
File Edit Selection View Go Run ... < > style
EXPLORER ... Css.jsx U abc.scss U App.js M
src > components > Css.jsx > External
16   return () =>
17     <h1 className="main">External style</h1>
18   ]
19 }
20 const ModuleCSS = () => {
21   return (
22     <h1 className={ButtonModule.btn}>Module style</h1>
23   )
24 }
25 const SassCSS = () => {
26   return (
27     <div className="sub">
28       <h1 className='sub_h1'>SaaS style</h1>
29       <h1 className='subAbc'>SaaS style</h1>
30     </div>
31   )
32 }
33 export { InlineCss, Stylesheet, External, ModuleCSS, SassCSS }
```



```
File Edit Selection View Go ... < > style
EXPLORER ... Css.jsx U abc.scss U App.js M
src > style > abc.scss > &.sub
1 .sub {
2   background-color: chartreuse;
3
4   /* & _h1 means a upar lakhela object nu name check
5    karse je hase teni sathe _h1 join kari dese
6    jena karane class banse .sub_h1 , te class ne je style
7    aapi hoy a lagi jase */
8   &_h1 {
9     color: red;
10    font-weight: 900;
11  }
12  &Abc {
13    background-color: pink;
14    color: blue;
15  }
16 }
```

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's
                    accordion body.</strong>
                    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.
                  </AccordionBody>
                </AccordionItem>
              <Fragment>
            )
          })
        }
      </div>
    
```

```
        </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;
```

The screenshot shows a Windows desktop environment. In the center is a Microsoft Edge browser window displaying a React application with the title 'reactstrap'. Below the browser is a code editor window showing the contents of 'App.js'. The code editor has tabs for 'Style.jsx', 'App.js', and 'package.json'. The 'App.js' tab is active, showing the following code:

```
src > App.js > ...
1 import 'bootstrap/dist/css/bootstrap.min.css';
2 import './App.css';
3 import Style from './components/Style';
4
5 function App() {
6   return (
7     <div className="App">
8       <Style />
9     </div>
10 );
11 }
12
13 export default App;
14
```

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>
)
}

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="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRh002iuK6FUUVVM"
  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>
    </form>
  </div>
</body>
```

```

                </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="angular" 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-
geWF76RCwLtnZ8qwWowPQNgL3RmwHVBC9FhGd1KrxdiJJigb/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 nicondition 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})">delete</button>
            </td>
        </tr>`
    })
    html += str
}

```

```

        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 StateManagement

```
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('javascrip
t')}>
    />
    <Label
        check
        for="exampleCheck2"
        className="px-2"
    >
        JavaScript
    </Label>
</div>
```

```
        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'>Employee 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>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>1</td>
                            <td><img alt='Profile Picture' src='https://via.placeholder.com/150x150' style='width: 100%; height: 100%;'/></td>
                            <td>John Doe</td>
                            <td>johndoe@example.com</td>
                            <td>johndoe@sample.com</td>
                            <td>1234567890</td>
                        </tr>
                    </tbody>
                </Table>
            </div>
        </Fragment>
    )
}

```

```

        </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;
```

The screenshot shows a VS Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "PASS-CHILD-TO-PARENT". The "src" folder contains "components" which has "ChildComp.jsx" and "ParentComp.jsx". Other files in "src" include "App.css", "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", "setupTests.js", ".gitignore", "package-lock.json", "package.json", and "README.md".
- Code Editor (Center):** The active file is "ParentComp.jsx". The code uses the useState hook to manage state and pass a function as a prop to the ChildComp component.
- Bottom Status Bar:** Shows the date and time (9:11 AM, 7/7/2023) and system icons.

```
src > components > ParentComp.jsx > ParentComp
1 import React, { useState } from 'react'
2 import ChildComp from './ChildComp'

3
4 function ParentComp() {
5   const [number, setnumber] = useState(1234)
6   /* to take data from child to parent , we use function which we define in parent component and
7   pass through props in child component and call in child component with argument which we want
8   in parent component and then use that from function parameter */
9   const dataFun = (a) => {
10     setnumber(a)
11   }
12   return (
13     <div>
14       <ChildComp data={dataFun} />
15     </div>
16   )
17 }

18 export default ParentComp
19
```

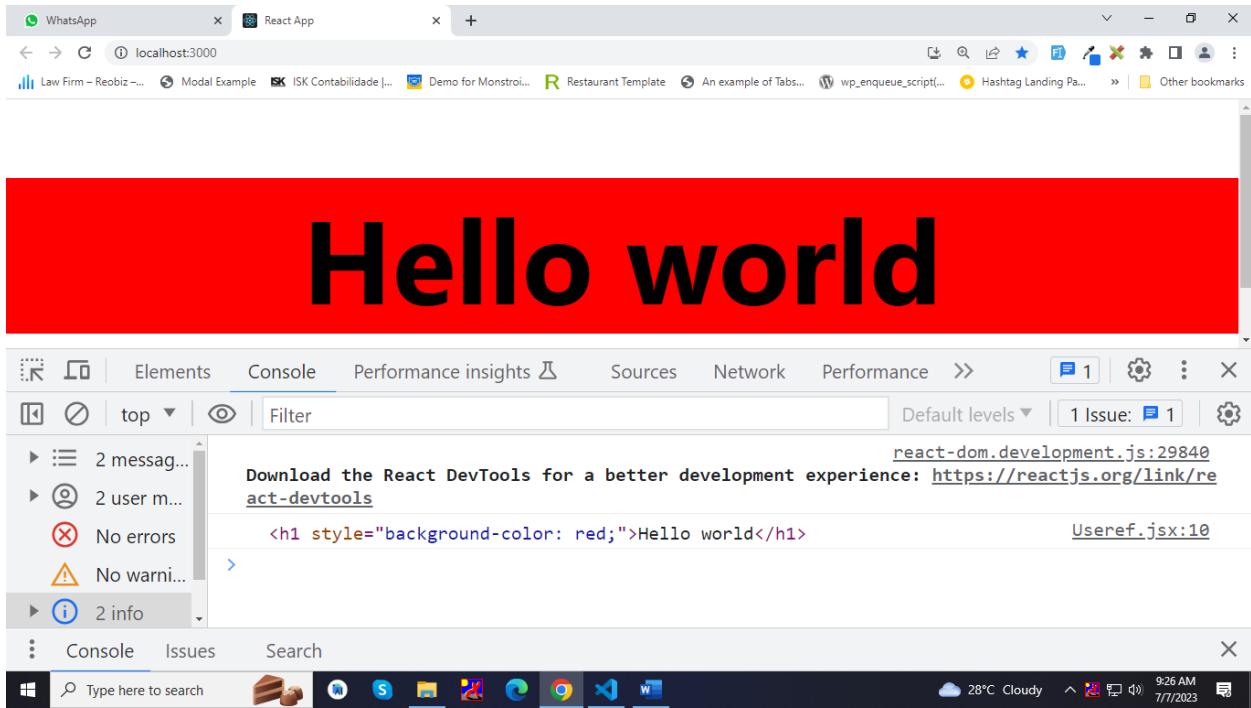
Userref

```
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



Higher Order Component

HOC

```
import { useRef } from "react"

export const HOC = (Component) => {
    /* HOC use to make common section use in all pages
       for that we make HOC function and then make component in that function in
       which we make common section and another component which show in that part like
       header or sidebar which show in all page that make and which part change in all
       page that part we take from component
       if we not make newcomponent in HOC then we can use Component facility like
       userref , useState etc.. */
    const NewComponent = () => {
        const reference = useRef()
        return (
            <div className="row m-0 g-0" style={{ height: '100vh' }}>
                <div className="col-2 bg-body-secondary">
                    <h1 className="p-2">Side Bar</h1>
```

```
        <h2><a href="#home">Home</a></h2>
        <h2><a href="#about">About</a></h2>
    </div>
    <div className="col-10">
        <header className="bg-info-subtle p-2">
            <h1 ref={reference}>Header</h1>
        </header>
        <Component reference={reference} />
    </div>
</div>
)
}
return NewComponent
}
```

Home.jsx

```
import React, { useEffect } from 'react'
import { HOC } from './HOC'

function Home(props) {
    useEffect(() => {
        console.log(props.reference)
    })
    return (
        <div>
            <h2 className='p-2 bg-primary' id="home">Home</h2>
        </div>
    )
}

export default HOC(Home)
```

About.jsx

```
import React, { Component } from 'react'
import { HOC } from './HOC'
```

```
class About extends Component {
  render() {
    return (
      <h2 className='p-2 bg-primary' id="about">About</h2>
    )
  }
}
export default HOC(About)
```

Use Context

App.js

```
/* Use Context is use to send data to any nested component. If we want to send
data from first component to nested end component it is very useful. we can send
data as a provider as a component and recieve by two method first is
context.consumer and other is useContext*/
import './App.css';
import First from './components/First';
import { createContext } from 'react';
export const MehulContext = createContext()

function App() {
  return (
    <div className="App">
      <MehulContext.Provider value={{ number: 10 }}>
        <First />
      </MehulContext.Provider>
    </div>
  );
}

export default App;
```

First.jsx

```
import React from 'react'
import Second from './Second'

function First() {
```

```
return (
  <>
    <h1>First</h1>
    <Second/>
  </>
)
}

export default First
```

Second.jsx

```
import React from 'react'
import Third from './Third'

function Second() {
  return (
    <>
      <h1>Second</h1>
      <Third />
    </>
  )
}

export default Second
```

Third.jsx

```
import { MehulContext } from '../App'

/* Use Context is use to send data to any nested component. If we want to send
data from first component to nested end component it is very useful. we can send
data as a provider as a component and recieve by two method first is
context.consumer and other is useContext*/
function Third() {
  /* use context Method 2 */
  // const value = useContext(MehulContext)
  return (
    <>
      /* use context Method 2 */
      /* <h1>{value.number}</h1> */
      /* use context Method 1 */
    </>
  )
}
```

```

        <MehulContext.Consumer>
            {value => { return <h1>{value.number}</h1> }}
        </MehulContext.Consumer>
    </>
)
}

export default Third

```

UseRef In Class Component

```

import React, { Component } from 'react'

export default class Form extends Component {
    constructor() {
        super()
        this.state = {
            href: {}
        }
    }
    /* when we want element in class component we can access using ref attribute,
when function call on change that file we get in that state which we define in
ref. and then if we want to reset file choose then do like below */
    getData = () => {
        console.log(this.state.href)
        // console.log(this.state.href.current.value = '')
    }
    render() {
        return (
            <div>
                <input type="file" onChange={this.getData} ref={this.state.href}>
            />
            </div>
        )
    }
}

```

Class Crud

Form Component

```
import React, { Component, Fragment, useContext } from 'react'
import { Button, Col, Container, Form, FormGroup, Input, Label, Row } from
'reactstrap'
import TableComponent from './TableComponent'

class FormComponent extends Component {
  constructor(props) {
    super(props)
    this.state = {
      obj: {},
      array: [],
      count: 0,
      editobj: {},
      FileRef: {}
    }
  }
  inputChange = async (e) => {
    if (e.target.name === "hobby") {
      if (e.target.checked === true) {
        this.state.obj.hobby = [
          ...this.state.obj.hobby ? [...this.state.obj.hobby] : [],
          e.target.value
        ]
      }
      else {
        this.state.obj.hobby = this.state.obj.hobby.filter((x) => {
          return x !== e.target.value
        })
      }
    }
    else if (e.target.name === "profile") {
      this.state.profile = await this.toBase64(e.target.files[0])
      this.state.obj = {
        ...this.state.obj, [e.target.name]: this.state.profile
      }
    }
    else {
      this.state.obj = {
        ...this.state.obj,
        [e.target.name]: e.target.value
      }
    }
  }
}

export default FormComponent
```

```

        }
    }
    this.setState({
        obj: this.state.obj
    })
}
toBase64 = (file) => {
    return new Promise((resolve, reject) => {
        const reader = new FileReader();
        reader.readAsDataURL(file);
        reader.onload = () => resolve(reader.result);
        reader.onerror = error => reject(error);
    })
}
submitFunction = (e) => {
    console.log(this.state.array)
    e.preventDefault()
    if (this.state.obj.id === undefined) {
        this.state.count++
        this.state.obj.id = this.state.count
        this.state.array?.push(this.state.obj)
    }
    else {
        this.state.array.splice(this.state.array.findIndex((x) => x.id ===
this.state.obj.id), 1, this.state.obj)
    }
    this.setState({
        array: this.state.array,
        obj: {}
    })
    localStorage.setItem('array', JSON.stringify(this.state.array))
    localStorage.setItem('count', JSON.stringify(this.state.count))
    // this.props.reference = ''
    // this.props.reference.current.value= "";
    console.log(this.state.FileRef.current)
    this.state.FileRef.current.value = ""
}
editFun = (id) => {
    this.state.obj = this.state.array.find((x) => x.id === id)
    this.setState({ obj: this.state.obj })
}
deleteFun = (id) => {
    this.state.count--
    this.state.array.splice(this.state.array.findIndex((x) => x.id === id),
1)
}

```

```

        this.setState({
            array: this.state.array
        })
        localStorage.setItem('array', JSON.stringify(this.state.array) || [])
        localStorage.setItem('count', JSON.stringify(this.state.count) || 0)
    }
    static getDerivedStateFromProps() {
        return {
            array: JSON.parse(localStorage.getItem('array')) || [],
            count: JSON.parse(localStorage.getItem('count')) || 0
        }
    }
    render() {
        return (
            <Fragment>
                <Container className='my-5'>
                    <Form className='border border-1 rounded-2 py-3'
onSubmit={this.submitFunction}>
                        <h1 className='text-center'>Employee Form</h1>
                        <Row>
                            <Col xs={6} className='offset-3 border border-1
rounded-2 py-3'>
                                <Row>
                                    <Col xs={12}>
                                        <FormGroup floating>
                                            <Input
                                                id="userName"
                                                name="userName"
                                                placeholder="User Name"
                                                type="text"
                                                value={this.state.obj.userName ?
this.state.obj.userName : ''}
                                                onChange={this.inputChange}
                                            />
                                            <Label for="userName">
                                                User Name
                                            </Label>
                                        </FormGroup>
                                    </Col>
                                    <Col xs={12}>
                                        <FormGroup floating>
                                            <Input
                                                id="email"
                                                name="email"
                                                placeholder="email"
                                            />
                                            <Label for="email">
                                                Email
                                            </Label>
                                        </FormGroup>
                                    </Col>
                                </Row>
                            </Col>
                        </Row>
                    </Form>
                </Container>
            </Fragment>
        )
    }
}

```

```
        type="email"
        value={this.state.obj.email ?  
this.state.obj.email : ''}
        onChange={this.inputChange}
    />
    <Label for="email">
        Email
    </Label>
</FormGroup>
</Col>
<Col xs={6}>
    <FormGroup floating>
        <Input
            id="phoneno"
            name="phoneno"
            placeholder=" Phone No"
            type="number"
            value={this.state.obj.phoneno ?  
this.state.obj.phoneno : ''}
            onChange={this.inputChange}
        />
        <Label for="phoneno">
            Phone No
        </Label>
    </FormGroup>
</Col>
<Col xs={6}>
    <FormGroup floating>
        <Input
            id="pincode"
            name="pincode"
            placeholder="Pin Code"
            type="number"
            value={this.state.obj.pincode ?  
this.state.obj.pincode : ''}
            onChange={this.inputChange}
        />
        <Label for="pincode">
            Pin Code
        </Label>
    </FormGroup>
</Col>
<Col xs={8}>
    <Label for="hobbies">
        Hobbies
    </Label>

```

```
</Label>
<div>
  <Input
    id="programming"
    name="hobby"
    type="checkbox"
    onChange={this.inputChange}
    value='programming'
    checked={this.state.obj.hobby?.in
cludes("programming") ? true : false}>
    />
    <Label for="programming">
      Programming
    </Label>
    <Input
      id="travelling"
      name="hobby"
      type="checkbox"
      onChange={this.inputChange}
      value="travelling"
      checked={this.state.obj.hobby?.in
cludes("travelling") ? true : false}>
      />
      <Label for="travelling">
        Travelling
      </Label>
      <Input
        id="dancing"
        name="hobby"
        type="checkbox"
        onChange={this.inputChange}
        value="dancing"
        checked={this.state.obj.hobby?.in
cludes("dancing") ? true : false}>
        />
        <Label for="dancing" className='px-
2'>
          Dancing
        </Label>
      </div>
    </Col>
    <Col xs={4}>
```

```
        <img src={this.state.obj.profile} alt="" style={{ width: "100%" }} />
      </Col>
      <Col xs={6}>
        <Label for="gender">
          Gender
        </Label>
        <div>
          <Input
            id="Male"
            name="gender"
            type="radio"
            value='Male'
            onChange={this.inputChange}
            checked={this.state.obj.gender
              === "Male"} />
          <Label for="Male" className='px-2'>
            Male
          </Label>
          <Input
            id="Female"
            name="gender"
            type="radio"
            value='Female'
            onChange={this.inputChange}
            checked={this.state.obj.gender
              === "Female"} />
          <Label for="Female" className='px-2'>
            Female
          </Label>
        </div>
      </Col>
      <Col xs={12}>
        <Label for="image" className='px-2'>
          Upload Image
        </Label>
        <input
          type='file'
          name='profile'
          id="image"
          onChange={this.inputChange}
          ref={this.state.FileRef}>
      </Col>
    </Row>
  </Form>

```

```
        </input>
    </Col>
    <Col className='text-center'>
        <Button>Submit</Button>
    </Col>
</Row>
</Col>
</Row>
</Form>
</Container>
<Container>
    <TableComponent array={this.state.array}
deleteFun={this.deleteFun} editFun={this.editFun} />
    </Container>
</Fragment >
)
}
}

export default FormComponent
```

Table.jsx

```
        </tr>
      </thead>
      <tbody>
        {this.props.array?.map((x, i) => {
          return (
            <tr key={i}>
              <td>{i + 1}</td>
              <td>
                <img src={x.profile} alt="" width={100}
height={100} />
              </td>
              <td>{x.userName}</td>
              <td>{x.email}</td>
              <td>{x.phoneno}</td>
              <td>{x.pincode}</td>
              <td>{x.gender}</td>
              <td>{x.hobby?.join(",")}</td>
              <td>
                <button className='me-2 px-2 text-white
bg-warning fs-5' onClick={() => { this.props.editFun(x.id) }}>Edit</button>
                <button className='text-bg-danger px-2
fs-5'
                      onClick={() => {
this.props.deleteFun(x.id) }}>Delete</button>
              </td>
            </tr>
          )
        )})
      </tbody>
    </Table>
  </Fragment>
)
}
}
export default TableComponent
```

Use Context Multiple Value

```
import React, { createContext, useState } from 'react'
import First from './First'
export const ReferenceContext = createContext()
function UseContextComponent() {
  const [arr, setarr] = useState(["hello", " ", "world"])
  return (
    <div>
      <ReferenceContext.Provider value={{ arr, setarr }}>
        <First />
      </ReferenceContext.Provider>
    </div>
  )
}
export default UseContextComponent
```

First.jsx

```
import React from 'react'
import { ReferenceContext } from './UseContextComponent'
import Second from './Second'

function First() {
  return (
    <div>
      <ReferenceContext.Consumer>
        {(value) => {
          return (
            <>
              <button onClick={() => {
                value.setarr(["Welcome", " ", "Guest"])
              }}>
                Click Me
              </button>
              {value.arr.join(' ')}
            </>
          )
        }}
      </ReferenceContext.Consumer>
    </div>
  )
}
export default First
```

```
                }]}>Change data</button>
            <h3><del>{value.arr}</del></h3>
        </>
    )
}
</ReferenceContext.Consumer>
<Second />
</div>
)
}

export default First
```

Second.jsx

```
import React, { useContext } from 'react'
import { ReferenceContext } from './UseContextComponent'
function Second() {
    const RefArray = useContext(ReferenceContext)
    return (
        <div>
            {console.log(RefArray.arr)}
            <h1>{RefArray.arr.join(" ")}</h1>
        </div>
    )
}

export default Second
```

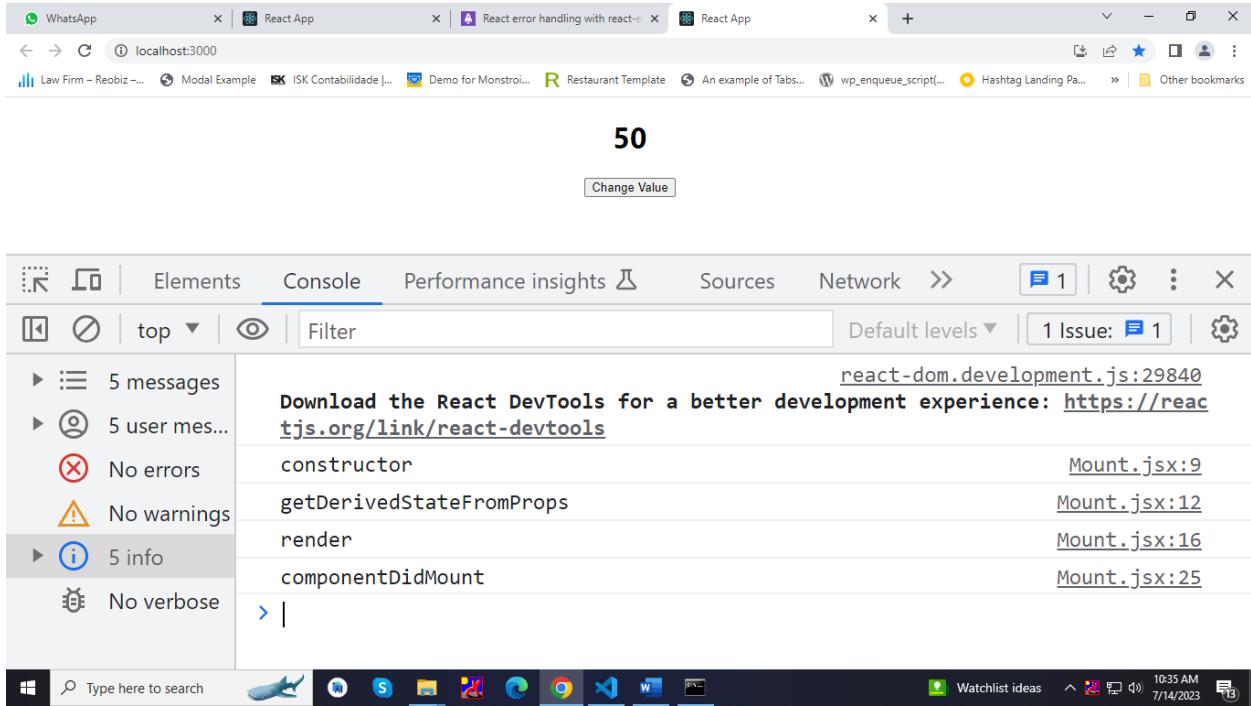
Class Life Cycle

Mounting

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

export default class Mount extends Component {
  constructor() {
    super()
    this.state = {
      number: 50
    }
    console.log("constructor")
  }
  static getDerivedStateFromProps() {
    console.log("getDerivedStateFromProps")
    return true
  }
  render() {
    console.log("render")
    return (
      <div>
        <h1>{this.state.number}</h1>
        <button onClick={() => this.setState({ number: this.state.number + 1 })}>Change Value</button>
      </div>
    )
  }
  componentDidMount() {
    console.log('componentDidMount')
  }
}
```

Ouput



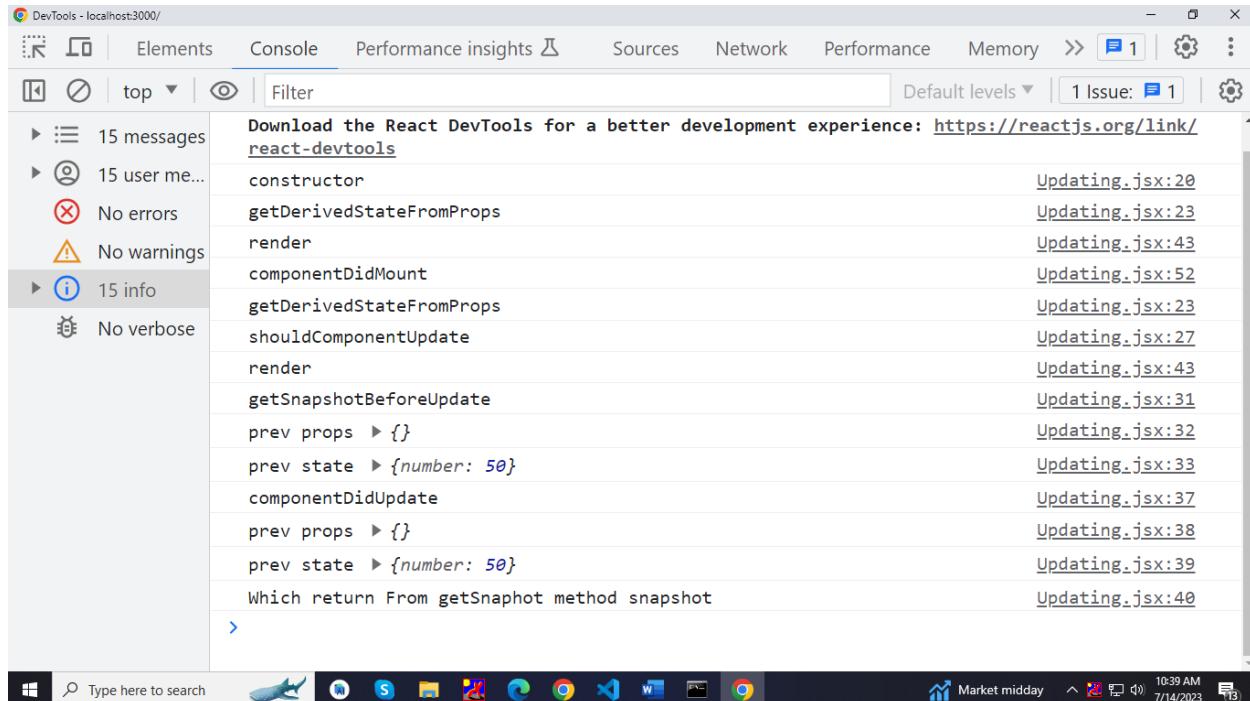
Updating

```
import React, { Component } from 'react'
/* Updating Phase =>5 Method
1=>getDerivedStateFromProps
2=>shouldComponentUpdate
3=>render
4=>getSnapshotBeforeUpdate
5=>componentDidUpdate

1 st and 3rd method is also part of updating phase which we use in Mounting phase
2=>ShouldComponentUpdate() is run before rerendering. If we return true then
render run if we return false then render not run
4=>getSnapshotBeforeUpdate() This method use for prevprops and prevstate and this
method run after rerender
5=>component did update method run after getsnapshot() in this methos same data
occur prevprops and prevstate and which we return from getsnapshotbeforeupdate
*/
export default class Updating extends Component {
  constructor() {
    super()
```

```
        this.state = {
          number: 50
        }
        console.log("constructor")
      }
      static getDerivedStateFromProps() {
        console.log("getDerivedStateFromProps")
        return true
      }
      shouldComponentUpdate() {
        console.log('shouldComponentUpdate')
        return true
      }
      getSnapshotBeforeUpdate(prevprops, prevstate) {
        console.log('getSnapshotBeforeUpdate')
        console.log("prev props", prevprops)
        console.log("prev state", prevstate)
        return "snapshot"
      }
      componentDidUpdate(prevprops, prevstate, snapshot) {
        console.log('componentDidUpdate')
        console.log("prev props", prevprops)
        console.log("prev state", prevstate)
        console.log("Which return From getSnapshot method", snapshot)
      }
      render() {
        console.log("render")
        return (
          <div>
            <h1>{this.state.number}</h1>
            <button onClick={() => this.setState({ number: this.state.number + 1 })}>Change Value</button>
          </div>
        )
      }
      componentDidMount() {
        console.log('componentDidMount')
      }
    }
  }
```

Output



Unmounting

Unmounting.jsx

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

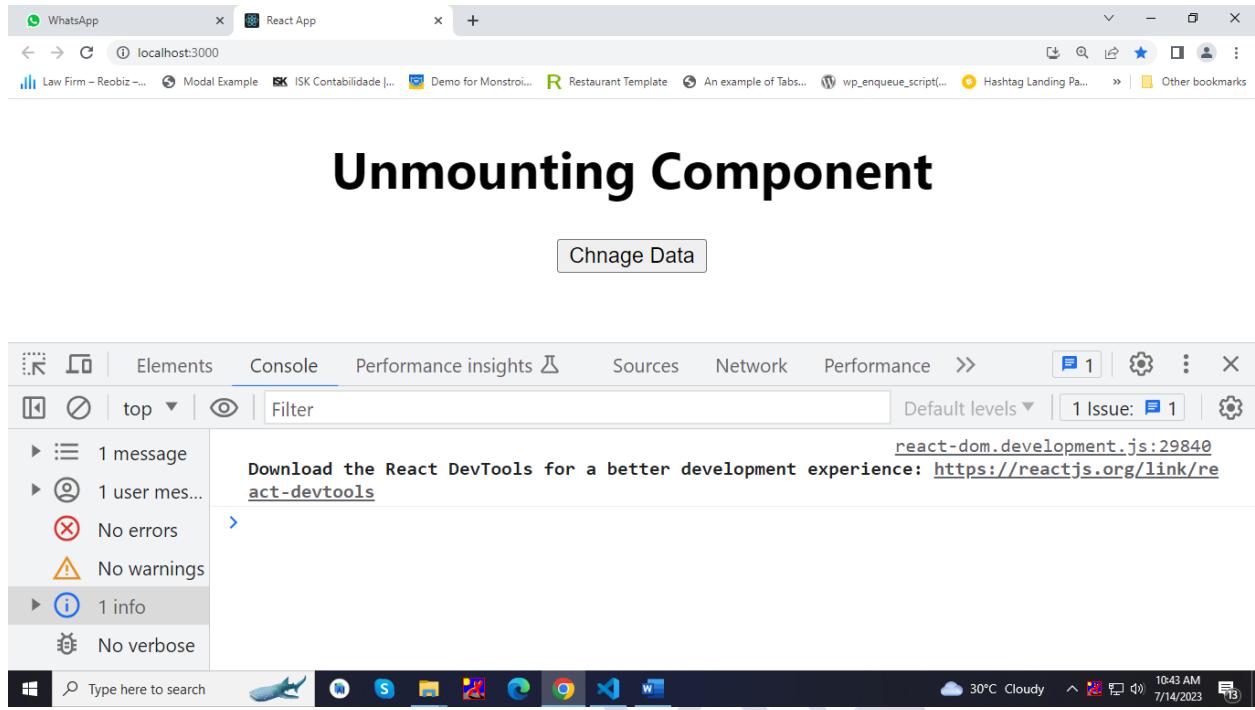
class Unmounting extends Component {
    /* this method call when component is removed from dom */
    componentWillUnmount() {
        console.log('componentWillUnmount')
    }
    render() {
        return (
            <div>
                <h1>Unmounting Component</h1>
            </div>
        )
    }
}
export default Unmounting
```

Main.jsx

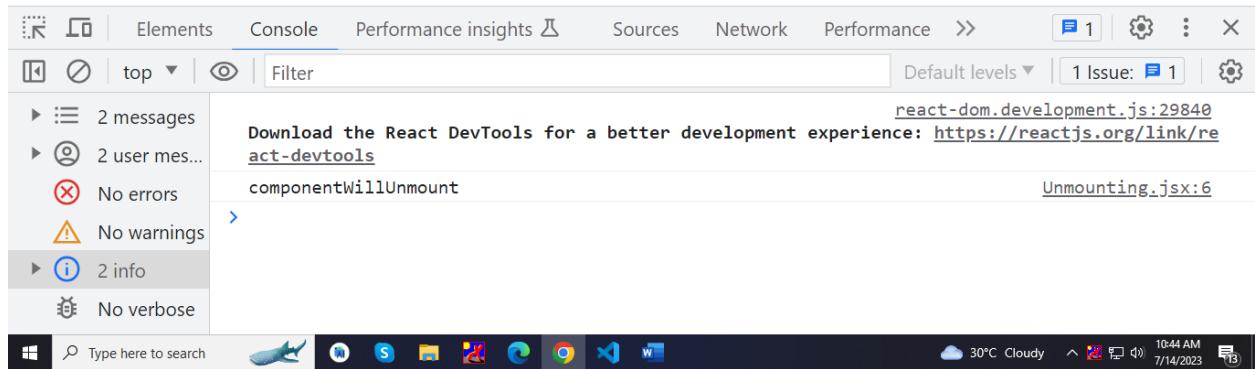
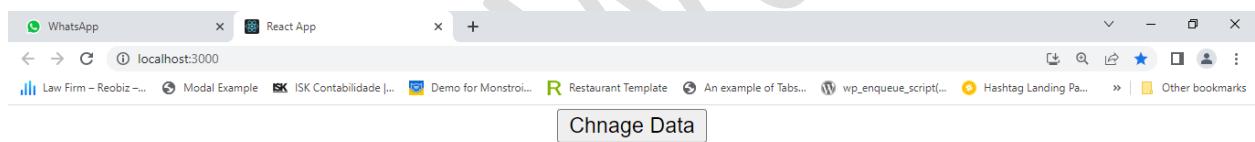
```
import React, { Component } from 'react'
import Unmounting from './Unmounting'

class Main extends Component {
  constructor() {
    super()
    this.state = {
      number: true
    }
  }
  render() {
    return (
      <div>
        {this.state.number === true && <Unmounting />}
        <button onClick={() => this.setState({ number: !this.state.number })}>Change Data</button>
      </div>
    )
  }
}
export default Main
```

Before unmount



After Unmount



Error Boundries

Error Component

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

export default class ErrorBoundries extends Component {
  constructor(props) {
    super(props)
    this.state = {
      error: false,
      data: '',
      info: ''
    }
  }
  /* A method jyare error aave tyare call thase ane parameter ma eror nu name aapse */
  static getDerivedStateFromError(err) {
    console.log(err)
    return { error: true }
  }
  /* A method jyare error aave tyare call thase ane ana parameter ma error nu name ane ani info aapse */
  componentDidCatch(errordata, errorInfo) {
    console.log(errordata, errorInfo);
    this.setState({
      data: errordata,
      info: errorInfo
    })
  }
  render() {
    if (this.state.error) {
      return (<>
        <h1>Something Went Wrong</h1>
        <h1>{this.state.data.toString()}</h1>
        <h1>{this.state.info.componentStack}</h1>
      </>)
    }
    else {
      return this.props.children
    }
  }
}
```

Child Component

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

export default class ChildComponent extends Component {
  constructor(props) {
    super(props)
    this.state = {

    }
    if (this.props.name === 'sagar') {
      throw new Error("oops Error occurred")
    }
  }

  render() {
    return (
      <div>
        <h1>
          ChildComponent
        </h1>
      </div>
    )
  }
}
```

App.js

```
import './App.css';
import ErrorBoundries from './components/ErrorBoundries';
import ChildComponent from './components/ChildComponent';

function App() {
  return (
    <div className="App">
      <ErrorBoundries>
        <ChildComponent name="mehul" />
      </ErrorBoundries>
      <ErrorBoundries>
        <ChildComponent name="pratik" />
      </ErrorBoundries>
      <ErrorBoundries>
        <ChildComponent name="sagar" />
      </ErrorBoundries>
    </div>
  );
}

export default App;
```

```
      </ErrorBoundries>
    </div>
  );
}

export default App;
```

Filter Function

Form.jsx

```
import { Fragment, useEffect, useState } from "react";
import { Col, Container, Form, FormGroup, Input, Label, Row } from "reactstrap";
import MainTable from "./MainTable";
import { AiOutlineArrowDown, AiOutlineArrowUp } from "react-icons/ai";
const MainForm = () => {
  let [count, setcount] = useState(1);
  const [regex, setregex] = useState(/^[A-Za-z]+$/);
  let [obj, setobj] = useState({
    username: "",
    lastname: "",
    email: "",
    password: "",
    confirmpassword: "",
    gender: "",
    phoneno: "",
    date: "",
    information: "",
    language: []
  });
  let [editobj, seteditobj] = useState({
    language: []
  });
  let [array, setarray] = useState([]);
  let [iserror, setiserror] = useState({});
  let [formHeight, setformHeight] = useState(false);
  let [filterObj, setFilterObj] = useState({
    filterFname: "",
    filterLName: "",
    filterEmail: ""
  });
}
```

```

let [filterArray, setfilterArray] = useState([]);

/* to capitalize first letter of each word */
const capitalize = (str) =>
  str
    .toLowerCase()
    .replace(/\w{1,}/g, (match) =>
      match.replace(/\w/, (m) => m.toUpperCase())
    );

const mainData = (e) => {
  if (e.target.name === "language") {
    if (e.target.checked) {
      obj.language.push(e.target.value);
    } else {
      obj.language = obj.language?.filter((x, i) => x !== e.target.value);
    }
  } else if (e.target.name === "information") {
    obj[e.target.name] = capitalize(e.target.value);
  } else {
    obj[e.target.name] = e.target.value;
  }

  setobj({ ...obj });
  validate(e.target.name);
};

const dateFunction = (year, month, day) => {
  return new Date(year + 18, month, day) <= new Date();
};

const validate = (name) => {
  /* ===== For User Name===== */
  if (name === "username") {
    if (!obj.username || obj.username === "") {
      iserror.username = "FirstName Should Not be empty";
    } else if (obj.username && !regex.test(obj.username)) {
      iserror.username = "FirstName should contains only alphabet";
    } else {
      delete iserror.username;
    }
  }
  /* ===== For User Name===== */
  if (name === "lastname") {
    if (!obj.lastname || obj.lastname === "") {
      iserror.lastname = "LastName Should Not be empty";
    }
  }
};

```

```

} else if (obj.lastname && !regex.test(obj.lastname)) {
    iserror.lastname = "lastName should contains only alphabet";
} else {
    delete iserror.lastname;
}
/*
=====
For Email=====
*/
if (name == "email") {
    if (!obj.email || obj.email === "") {
        iserror.email = "Email Should Not be empty";
    } else if (
        obj.email &&
        !obj.email.includes("@gmail.com") &&
        !obj.email.includes("@outlook.com") &&
        !obj.email.includes("@mailinator.com")
    ) {
        iserror.email = "Email Should include @gmail or @outlook.com";
    } else {
        delete iserror.email;
    }
}
/*
=====
For Date =====
*/
if (name == "date") {
    let date = new Date(obj.date);
    if (!obj.date || obj.date === "") {
        iserror.date = "date Should Not be empty";
    } else if (
        !dateFunction(date.getFullYear(), date.getMonth(), date.getDate())
    ) {
        iserror.date = "Age Should be 18+";
    } else {
        delete iserror.date;
    }
}

// else if (
//     obj.date &&
//     (new Date().getTime() - new Date(obj.date).getTime()) /
//     (1000 * 3600 * 24 * 365) <=
//     18
// ) {
//     iserror.date = "Age Should be 18+";
// }

/*
=====
For Password =====
*/

```

```

/*===== For Confirm Password =====*/
if (name == "password" || name == "confirmpassword") {
    if (!obj.password || obj.password === "") {
        iserror.password = "Password Should Not Be Empty";
    } else if (
        obj.password &&
        !/^([a-z]+[A-Z]+[0-9]+[@$!%*?&])[A-Za-z\d@#$!%*?&]{8,}$/.test(
            obj.password
        )
    ) {
        iserror.password =
            "Minimum eight,at least one uppercase,lowercase letter,one number and
one special character:";
    } else {
        delete iserror.password;
    }
    if (!obj.confirmpassword || obj.confirmpassword === "") {
        iserror.confirmpassword = "Confirm Password Should Not Be Empty";
    } else if (!(obj.confirmpassword === obj.password)) {
        iserror.confirmpassword =
            "Password and Confirm Password Should Be Same";
    } else {
        delete iserror.confirmpassword;
    }
}

/*===== For Mobile Number =====*/
if (name == "phoneno") {
    if (!obj.phoneno || obj.phoneno === "") {
        iserror.phoneno = "Mobile number should not be empty";
    } else if (
        obj.phoneno &&
        (obj.phoneno <= 0 || obj.phoneno.length != 10)
    ) {
        iserror.phoneno = "Please Enter valid Mobile No";
    } else {
        delete iserror.phoneno;
    }
}
/*===== For Gender =====*/
if (name == "gender") {
    if (!obj.gender) {
        iserror.gender = "Gender is Required";
    } else {

```

```

        delete iserror.gender;
    }
}

/*===== For Language ===== */
if (name == "language") {
    if (!obj.language || obj.language.length < 3) {
        iserror.language = "Please Select atleast Three";
    } else {
        delete iserror.language;
    }
}
/*===== For Information ===== */
if (name == "information") {
    if (!obj.information || obj.information === "") {
        iserror.information = "Information is required";
    } else {
        delete iserror.information;
    }
}
setiserror({ ...iserror });
};

const submitFunction = (e) => {
e.preventDefault();
/* to send all key for validation when form is blank and we click submit */
for (let key in obj) {
    validate(key);
}
/* if we fill all field true in form all object key is delete and iserror
object become blank. to check blank we convert array key in to array as below. we
can not compare blank object with blank object*/
if (Object.keys(iserror).length == 0) {
    if (obj.id === undefined) {
        count++;
        obj["id"] = count;
        array = [...array, obj];

        setcount(count);
    } else {
        array.splice(
            array.findIndex((x) => x.id === obj.id),
            1,
            obj
        );
    }
}

```

```

        }
    }
    localStorage.setItem("array", JSON.stringify(array));
    localStorage.setItem("count", count);
    obj = {
        username: "",
        lastname: "",
        email: "",
        phoneno: "",
        pincode: "",
        gender: "",
        language: [],
    };
    setobj({ ...obj });
    setarray([...array]);
    filterArray = array;
    setfilterArray([...filterArray]);
    filterFunction(1);
};

function editfun(id) {
    editobj = array.find((x) => x.id === id);
    setobj({ ...editobj });
}
function deletefun(id) {
    array.splice(
        array.findIndex((x) => x.id === id),
        1
    );
    filterArray = array;
    setarray([...array]);
    setfilterArray([...filterArray]);

    localStorage.setItem("array", JSON.stringify(array));
    filterFunction(1);
}

useEffect(() => {
    setarray(JSON.parse(localStorage.getItem("array")) || []);
    setfilterArray(JSON.parse(localStorage.getItem("array")) || []);
    setcount(JSON.parse(localStorage.getItem("count")) || 0);
}, []);
const formOpen = () => {
    formHeight = true;
    setformHeight(formHeight);
}

```

```
};

/* when we want to form close we set height in which only form heading show and
style overflow hidden */
const formClose = () => {
  formHeight = false;
  setformHeight(formHeight);
};

const filterFunction = (e) => {
  /* jyare delete edit ke submit kariye tyare pan filter thatu rakhva mate
filterfunction call karyu che submit na function par pan tyare aapri pase koi key
hoti nathi ane aapne key ni jarur nathi pan key nai male to e.target.name ma
error aape atle aapre tya one pass karyo che so jo e ma one aave to upar ni 2
line nu work stop thai jase */
  if (e != 1) {
    filterObj[e.target.name] = e.target.value;
    setFilterObj({ ...filterObj });
  }
  filterArray = array.filter(
    (x) =>
      /* if x.lastname or any other data is not available and we give &&
condition between them then it is not check that undefined data condition or skip
it therefore if any one field of filter we fill then only that condition check
other is skip and filter only according to one condition. when filter by two data
then two condition and another one skip nad so on */
      x.username
        .toLowerCase()
        .includes(filterObj.filterFname.toLowerCase()) &&
      x.lastname
        .toLowerCase()
        .includes(filterObj.filterLName.toLowerCase()) &&
      x.email.toLowerCase().includes(filterObj.filterEmail.toLowerCase())
  );
}

// filterArray = array.filter((x) => {
//   if (filterObj.filterFname?.length != 0 || filterObj.filterLName?.length
!= 0 || filterObj.filterEmail?.length != 0) {
//     if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0 && filterObj.filterEmail?.length != 0) {
//       if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
//         return x;
//       }
//     }
//   }
// }
```

```
//      else if (filterObj.filterLName?.length != 0 &&
filterObj.filterEmail?.length != 0) {
//          if (x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterFname?.length != 0 &&
filterObj.filterEmail?.length != 0) {
//          if (x.username.includes(filterObj.filterFname) &&
x.email.includes(filterObj.filterEmail)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0) {
//          if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterLName?.length != 0) {
//          if (x.lastname.includes(filterObj.filterLName)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterEmail?.length != 0) {
//          if (x.email.includes(filterObj.filterEmail)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterFname?.length != 0) {
//          if (x.username.includes(filterObj.filterFname)) {
//              return x;
//          }
//      }
//  }
//  else {
//      return x
//  }
// })
setfilterArray(filterArray);
};

return (
<Fragment>
```

```
<div className="container mt-2 mb-5 border border-1 border-black rounded p-2">
  <h1 className="text-center"> Form Filter </h1>
  <div className="row">
    <div className="col-4">
      <FormGroup>
        <Label for="filterFname" className="fw-600 fs-5">
          User Name
        </Label>
        <Input
          id="filterFname"
          name="filterFname"
          placeholder="User name"
          type="text"
          onChange={filterFunction}
        />
        <span className="text-danger">{iserror.email}</span>
      </FormGroup>
    </div>
    <div className="col-4">
      <FormGroup>
        <Label for="lastName" className="fw-600 fs-5">
          Last Name
        </Label>
        <Input
          id="lastName"
          name="filterLName"
          placeholder="Last Name"
          type="text"
          className="main"
          onChange={filterFunction}
        />
        <span className="text-danger">{iserror.email}</span>
      </FormGroup>
    </div>
    <div className="col-4">
      <FormGroup>
        <Label for="filterEmail" className="fw-600 fs-5">
          Email
        </Label>
        <Input
          id="filterEmail"
          name="filterEmail"
          placeholder="Filter email"
          type="email"
        />
      </FormGroup>
    </div>
  </div>
</div>
```

```
        onChange={filterFunction}
      />
      <span className="text-danger">{iserror.email}</span>
    </FormGroup>
  </div>
</div>
</div>
<Row>
  <Col xs={6} className="offset-3">
    <Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
      <h1 className="text-center py-3">
        Employee Form{" "}
        {formHeight ? (
          <AiOutlineArrowUp onClick={formClose} />
        ) : (
          <AiOutlineArrowDown onClick={formOpen} />
        )}
      </h1>
      <Form
        onSubmit={(e) => {
          submitFunction(e);
        }}
        style={{
          height: formHeight ? "auto" : "0px",
          overflow: "hidden",
        }}
      >
        <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}
              />
              <span className="text-danger">{iserror.username}</span>
            </FormGroup>
```

```
</Col>
<Col md={6}>
  <FormGroup>
    <Label for="lastname" className="fw-600 fs-5">
      Last Name
    </Label>
    <Input
      id="lastname"
      name="lastname"
      placeholder=""
      type="text"
      className="main"
      value={obj.lastname || ""}
      onChange={mainData}
    />
    <span className="text-danger">{iserror.username}</span>
  </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}
    />
    <span className="text-danger">{iserror.email}</span>
  </FormGroup>
</Col>
<Col md={6}>
  <FormGroup>
    <Label for="date" className="fw-600 fs-5">
      Birth Date
    </Label>
    <Input
      id="date"
      name="date"
      placeholder=""
      type="date"
    />
```

```
        value={obj.date || ""}
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">{iserror.date}</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="password" className="fw-600 fs-5">
        Password
    </Label>
    <Input
        id="password"
        name="password"
        placeholder=""
        type="password"
        value={obj.password || ""}
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">{iserror.password}</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="confirmpassword" className="fw-600 fs-5">
        Confirm Password
    </Label>
    <Input
        id="confirmpassword"
        name="confirmpassword"
        placeholder=""
        type="password"
        value={obj.confirmpassword || ""}
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">
        {iserror.confirmpassword}
    </span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
```

```
<Label for="phoneno" className="fw-600 fs-5">
  Mobile no
</Label>
<Input
  id="phoneno"
  name="phoneno"
  placeholder=""
  value={obj.phoneno || ""}
  type="number"
  className="main"
  onChange={mainData}
/>
<span className="text-danger">{iserror.phoneno}</span>
</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="exampleCheck4"
        name="language"
        type="checkbox"
        className="language me-2"
        value="react"
        onChange={mainData}
        checked={obj.language.includes("react")}

    />
    <Label check for="exampleCheck4" className="px-2">
        REACT
    </Label>
</div>
</div>
<span className="text-danger">{iserror.language}</span>
</Col>
<Col md={6}>
    <Label for="example" className="fw-600 fs-5">
        Gender
    </Label>
```

```
<div className="d-flex">
  <div>
    <Input
      id="exampleCheck5"
      name="gender"
      type="radio"
      className="gender me-2"
      value="male"
      onChange={mainData}
      checked={obj.gender === "male"}
    />
    <Label check for="exampleCheck5" 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>
  <span className="text-danger">{iserror.gender}</span>
</Col>
<Col md={12}>
  <FormGroup>
    <Label for="information" className="fw-600 fs-5">
      Information
    </Label>
    <Input
      id="information"
      name="information"
      placeholder=""
      value={obj.information || ""}
      type="textarea"
      className="main"
      onChange={mainData}
    />
  </FormGroup>
</Col>
```

```

        />
        <span className="text-danger">{iserror.information}</span>
      </FormGroup>
    </Col>
  </Row>
  <div className="text-center">
    <button className="my-2 btn btn-secondary submit fs-4">
      Submit
    </button>
  </div>
</Form>
</Container>
</Col>
</Row>
<Container>
  <MainTable
    mainArray={array}
    filterArray={filterArray}
    editfun={editfun}
    deletefun={deletefun}
  />
</Container>
<Fragment>
);
};

export default MainForm;

```

Table.jsx

```

import React, { Fragment } from "react";
import { Table } from "reactstrap";

function MainTable(props) {
  return (
    <Fragment>
      <div className="my-5">
        <h1 className="text-center py-3">Employee Information</h1>
        <Table className="p-3">
          <thead>
            <tr>
              <th>Sr No</th>
              <th>Profile</th>
            </tr>
          </thead>

```

```

<th>Name</th>
<th>LastName</th>
<th>Email</th>
<th>Phone No</th>
<th>Gender</th>
<th>Language</th>
<th>Action</th>
</tr>
</thead>
<tbody>
    /* main array ni jagyae direct filter array par map chalavyo che so
jo koi filter check nai karyu hoy tyare badha undefine hase ane full array a
filter array ma return aavse ane a table ma dekhase ane jyare filter karsu tyare
to filter array j batavano che */
    {/* {console.log(props.filterArray)} */}
{props.filterArray.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.lastname}</td>
            <td>{x.email}</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;

```

Use Memo

```

import React, { useMemo, useState } from "react"

/* use Memo is same as useEffect but we can not use hook (useEffect) as a
variable but we can use Use Memo as a variable */
const UseMemo = () => {
    const [count, setcount] = useState(1)
    const Mehul = useMemo(() => {
        console.log(count)
        return count;
    }, [count])
    return (
        <>
            <h2>{Mehul}</h2>
            <h2>{count}</h2>
            <button onClick={() => setcount(count + 1)}>Click</button>
        </>
    )
}
export default UseMemo

```

Validation Using Json Data

Mainform.jsx

```

import { Fragment, useEffect, useState } from "react";
import { Col, Container, Form, FormGroup, Input, Label, Row } from "reactstrap";
import MainTable from "./MainTable";
import { AiOutlineArrowDown, AiOutlineArrowUp } from "react-icons/ai";
import ValidationData from '../components/validation/validation.json'
const MainForm = () => {
    let [count, setcount] =

```

```

const [regex, setregex] = useState(/^[A-Za-z]+$/);
let [obj, setobj] = useState({
    username: "",
    lastname: "",
    email: "",
    password: "",
    confirmpassword: "",
    gender: "",
    phoneno: "",
    date: "",
    information: "",
    language: []
});
let [editobj, seteditobj] = useState({
    language: []
});
let [array, setarray] = useState([]);
let [iserror, setiserror] = useState({});
let [formHeight, setformHeight] = useState(false);
let [filterObj, setFilterObj] = useState({
    filterFname: "",
    filterLName: "",
    filterEmail: ""
});
let [filterArray, setfilterArray] = useState([]);
/* to capitalize first letter of each word */
const capitalize = (str) =>
    str
        .toLowerCase()
        .replace(/\w{1,}/g, (match) =>
            match.replace(/\w/, (m) => m.toUpperCase())
        );
const mainData = (e) => {
    if (e.target.name === "language") {
        if (e.target.checked) {
            obj.language.push(e.target.value);
        } else {
            obj.language = obj.language?.filter((x, i) => x !== e.target.value);
        }
    } else if (e.target.name === "information") {
        obj[e.target.name] = capitalize(e.target.value);
    } else {
        obj[e.target.name] = e.target.value;
    }
}

```

```

    setobj({ ...obj });
    validate(e.target.name);
};

const dateFunction = (year, month, day) => {
    return new Date(year + 18, month, day) <= new Date();
};
const validate = (name) => {
    let pRegex = /^[?=.]*[a-z][?=.]*[A-Z][?=.]*\d[?=.]*[@$!%*?&][A-Za-zA\d@$!%*?&]{8,}$/;
    /* if we write iserror.name= 'djfshd' then it is take name as a property not
take as a variable
    and if we write iserror[name] then it takes name as variable and make
property according variable value

```

why we use find method in validation obj?

because find method find only one element of array and therefore when change event call that time in name only that field name occur and find method find only that obj.

why we use find method in errorobj?

because if we use for each method then it is check all the condition and we want if and else if type process

if we use find then if condition become true then it is not check any next conditions. But for that reason we have to set erroor condition according to order wise like first if then else if condition and so on*/

```

let validationObj = ValidationData.find(x => x.name === name)
console.log(validationObj)
let errorObj = validationObj.conditions.find((x) => eval(x.condition))
console.log(errorObj)
if (validationObj) {
    if (errorObj) {
        if (errorObj.otherField) {
            iserror[errorObj.otherField] = errorObj.error
        }
        else {
            iserror[name] = errorObj.error
        }
    }
    else {
        delete iserror[name]
    }
}
// /* ===== For User Name===== */

```

```
// if (name == "username") {
//   if (!obj.username || obj.username === "") {
//     iserror.username = "FirstName Should Not be empty";
//   } else if (obj.username && !regex.test(obj.username)) {
//     iserror.username = "FirstName should contains only alphabet";
//   } else {
//     delete iserror.username;
//   }
// }
// /* ===== For User Name===== */
// if (name == "lastname") {
//   if (!obj.lastname || obj.lastname === "") {
//     iserror.lastname = "LastName Should Not be empty";
//   } else if (obj.lastname && !regex.test(obj.lastname)) {
//     iserror.lastname = "lastName should contains only alphabet";
//   } else {
//     delete iserror.lastname;
//   }
// }
// /* ===== For Email===== */
// if (name == "email") {
//   if (!obj.email || obj.email === "") {
//     iserror.email = "Email Should Not be empty";
//   } else if (
//     obj.email &&
//     !obj.email.includes("@gmail.com") &&
//     !obj.email.includes("@outlook.com") &&
//     !obj.email.includes("@mailinator.com")
//   ) {
//     iserror.email = "Email Should include @gmail or @outlook.com";
//   } else {
//     delete iserror.email;
//   }
// }
// /* ===== For Date ===== */
// if (name == "date") {
//   let date = new Date(obj.date);
//   if (!obj.date || obj.date === "") {
//     iserror.date = "date Should Not be empty";
//   } else if (
//     !dateFunction(date.getFullYear(), date.getMonth(), date.getDate())
//   ) {
//     iserror.date = "Age Should be 18+";
//   } else {
//     delete iserror.date;
```

```

//      }
//    }

//    // else if (
//    //   obj.date &&
//    //   (new Date().getTime() - new Date(obj.date).getTime()) /
//    //   (1000 * 3600 * 24 * 365) <=
//    //   18
//    // ) {
//    //   iserror.date = "Age Should be 18+";
//    // }

//    /* ===== For Password ===== */
//    /*===== For Confirm Password ===== */
//    if (name == "password" || name == "confirmpassword") {
//      if (!obj.password || obj.password === "") {
//        iserror.password = "Password Should Not Be Empty";
//      } else if (
//        obj.password &&
//        !/^([a-z])([A-Z])(\d)([$!%*?&])[A-Za-
z\d@$!%*?&]{8,}$.test(
//          obj.password
//        )
//      ) {
//        iserror.password =
//          "Minimum eight,at least one uppercase,lowercase letter,one number
and one special character:";
//      } else {
//        delete iserror.password;
//      }
//      if (!obj.confirmpassword || obj.confirmpassword === "") {
//        iserror.confirmpassword = "Confirm Password Should Not Be Empty";
//      } else if (!(obj.confirmpassword === obj.password)) {
//        iserror.confirmpassword =
//          "Password and Confirm Password Should Be Same";
//      } else {
//        delete iserror.confirmpassword;
//      }
//    }

//    /*===== For Mobile Number =====*/
//    if (name == "phoneno") {
//      if (!obj.phoneno || obj.phoneno === "") {
//        iserror.phoneno = "Mobile number should not be empty";
//      } else if (

```

```

//      obj.phoneno &&
//      (obj.phoneno <= 0 || obj.phoneno.length != 10)
//    ) {
//      iserror.phoneno = "Please Enter valid Mobile No";
//    } else {
//      delete iserror.phoneno;
//    }
//  }
//  /*===== For Gender =====*/
//  if (name == "gender") {
//    if (!obj.gender) {
//      iserror.gender = "Gender is Required";
//    } else {
//      delete iserror.gender;
//    }
//  }

//  /*===== For Language =====*/
//  if (name == "language") {
//    if (!obj.language || obj.language.length < 3) {
//      iserror.language = "Please Select atleast Three";
//    } else {
//      delete iserror.language;
//    }
//  }
//  /*===== For Information =====*/
//  if (name == "information") {
//    if (!obj.information || obj.information === "") {
//      iserror.information = "Information is required";
//    } else {
//      delete iserror.information;
//    }
//  }
//  setiserror({ ...iserror });
};

const submitFunction = (e) => {
  e.preventDefault();
  /* to send all key for validation when form is blank and we click submit */
  for (let key in obj) {
    validate(key);
  }
  /* if we fill all field true in form all object key is delete and iserror
object become blank. to check blank we convert array key in to array as below. we
can not compare blank object with blank object*/

```

```
if (Object.keys(iserror).length == 0) {
  if (obj.id === undefined) {
    count++;
    obj["id"] = count;
    array = [...array, obj];

    setcount(count);
  } else {
    array.splice(
      array.findIndex((x) => x.id === obj.id),
      1,
      obj
    );
  }
}

localStorage.setItem("array", JSON.stringify(array));
localStorage.setItem("count", count);
obj = {
  username: "",
  lastname: "",
  email: "",
  phoneno: "",
  pincode: "",
  gender: "",
  language: [],
};
setobj({ ...obj });
setarray([...array]);
filterArray = array;
setfilterArray([...filterArray]);
filterFunction(1);
};

function editfun(id) {
  editobj = array.find((x) => x.id === id);
  setobj({ ...editobj });
}
function deletefun(id) {
  array.splice(
    array.findIndex((x) => x.id === id),
    1
  );
  filterArray = array;
  setarray([...array]);
  setfilterArray([...filterArray]);
}
```

```

localStorage.setItem("array", JSON.stringify(array));
filterFunction(1);
}

useEffect(() => {
  setarray(JSON.parse(localStorage.getItem("array")) || []);
  setfilterArray(JSON.parse(localStorage.getItem("array")) || []);
  setcount(JSON.parse(localStorage.getItem("count")) || 0);
}, []);

const formOpen = () => {
  formHeight = true;
  setformHeight(formHeight);
};

/* when we want to form close we set height in which only form heading show and
style overflow hidden */
const formClose = () => {
  formHeight = false;
  setformHeight(formHeight);
};

const filterFunction = (e) => {
  /* jyare delete edit ke submit kariye tyare pan filter thatu rakhva mate
filterfunction call karyu che submit na function par pan tyare aapri pase koi key
hoti nathi ane aapne key ni jarur nathi pan key nai male to e.target.name ma
error aape atle aapre tya one pass karyo che so jo e ma one aave to upar ni 2
line nu work stop thai jase */
  if (e != 1) {
    filterObj[e.target.name] = e.target.value;
    setFilterObj({ ...filterObj });
  }
  filterArray = array.filter(
    (x) =>
      /* if x.lastname or any other data is not available and we give &&
condition between them then it is not check that undefined data condition or skip
it therefore if any one field of filter we fill then only that condition check
other is skip and filter only according to one condition. when filter by two data
then two condition and another one skip nad so on */
      x.username
        .toLowerCase()
        .includes(filterObj.filterFname.toLowerCase()) &&
      x.lastname
        .toLowerCase()
        .includes(filterObj.filterLName.toLowerCase()) &&
      x.email.toLowerCase().includes(filterObj.filterEmail.toLowerCase())
  );
}

```

```
);

    // filterArray = array.filter((x) => {
    //   if (filterObj.filterFname?.length != 0 || filterObj.filterLName?.length
!= 0 || filterObj.filterEmail?.length != 0) {
    //     if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0 && filterObj.filterEmail?.length != 0) {
    //       if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
    //         return x;
    //       }
    //     }
    //     else if (filterObj.filterLName?.length != 0 &&
filterObj.filterEmail?.length != 0) {
    //       if (x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
    //         return x;
    //       }
    //     }
    //     else if (filterObj.filterFname?.length != 0 &&
filterObj.filterEmail?.length != 0) {
    //       if (x.username.includes(filterObj.filterFname) &&
x.email.includes(filterObj.filterEmail)) {
    //         return x;
    //       }
    //     }
    //     else if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0) {
    //       if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName)) {
    //         return x;
    //       }
    //     }
    //     else if (filterObj.filterLName?.length != 0) {
    //       if (x.lastname.includes(filterObj.filterLName)) {
    //         return x;
    //       }
    //     }
    //     else if (filterObj.filterEmail?.length != 0) {
    //       if (x.email.includes(filterObj.filterEmail)) {
    //         return x;
    //       }
    //     }
    //   }
    //   else if (filterObj.filterFname?.length != 0) {
```

```
//         if (x.username.includes(filterObj.filterFname)) {
//             return x;
//         }
//     }
// else {
//     return x
// }
// })
setfilterArray(filterArray);
};

return (
<Fragment>
<div className="container mt-2 mb-5 border border-1 border-black rounded p-2">
    <h1 className="text-center"> Form Filter </h1>
    <div className="row">
        <div className="col-4">
            <FormGroup>
                <Label htmlFor="filterFname" className="fw-600 fs-5">
                    User Name
                </Label>
                <Input
                    id="filterFname"
                    name="filterFname"
                    placeholder="User name"
                    type="text"
                    onChange={filterFunction}
                />
                <span className="text-danger">{iserror.email}</span>
            </FormGroup>
        </div>
        <div className="col-4">
            <FormGroup>
                <Label htmlFor="lastName" className="fw-600 fs-5">
                    Last Name
                </Label>
                <Input
                    id="lastName"
                    name="filterLName"
                    placeholder="Last Name"
                    type="text"
                    className="main"
                    onChange={filterFunction}
                />
            </FormGroup>
        </div>
    </div>
</div>
```

```
        <span className="text-danger">{iserror.email}</span>
      </FormGroup>
    </div>
    <div className="col-4">
      <FormGroup>
        <Label for="filterEmail" className="fw-600 fs-5">
          Email
        </Label>
        <Input
          id="filterEmail"
          name="filterEmail"
          placeholder="Filter email"
          type="email"
          onChange={filterFunction}
        />
        <span className="text-danger">{iserror.email}</span>
      </FormGroup>
    </div>
  </div>
<Row>
  <Col xs={6} className="offset-3">
    <Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
      <h1 className="text-center py-3">
        Employee Form{" "}
        {formHeight ? (
          <AiOutlineArrowUp onClick={formClose} />
        ) : (
          <AiOutlineArrowDown onClick={formOpen} />
        )}
      </h1>
      <Form
        onSubmit={(e) => {
          submitFunction(e);
        }}
        style={{
          height: formHeight ? "auto" : "0px",
          overflow: "hidden",
        }}
      >
        <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}
    />
    <span className="text-danger">{iserror.username}</span>
</FormGroup>
</Col>
<Col md={6}>
    <FormGroup>
        <Label for="lastname" className="fw-600 fs-5">
            Last Name
        </Label>
        <Input
            id="lastname"
            name="lastname"
            placeholder=""
            type="text"
            className="main"
            value={obj.lastname || ""}
            onChange={mainData}
        />
        <span className="text-danger">{iserror.lastname}</span>
    </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}
        />
```

```
        <span className="text-danger">{iserror.email}</span>
      </FormGroup>
    </Col>
    <Col md={6}>
      <FormGroup>
        <Label for="date" className="fw-600 fs-5">
          Birth Date
        </Label>
        <Input
          id="date"
          name="date"
          placeholder=""
          type="date"
          value={obj.date || ""}
          className="main"
          onChange={mainData}
        />
        <span className="text-danger">{iserror.date}</span>
      </FormGroup>
    </Col>
    <Col md={6}>
      <FormGroup>
        <Label for="password" className="fw-600 fs-5">
          Password
        </Label>
        <Input
          id="password"
          name="password"
          placeholder=""
          type="password"
          value={obj.password || ""}
          className="main"
          onChange={mainData}
        />
        <span className="text-danger">{iserror.password}</span>
      </FormGroup>
    </Col>
    <Col md={6}>
      <FormGroup>
        <Label for="confirmpassword" className="fw-600 fs-5">
          Confirm Password
        </Label>
        <Input
          id="confirmpassword"
          name="confirmpassword"
        />
      </FormGroup>
    </Col>
  </Row>

```

```
placeholder=""
type="password"
value={obj.confirmppassword || ""}
className="main"
onChange={mainData}
/>
<span className="text-danger">
  {iserror.confirmppassword}
</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
  <Label for="phoneno" className="fw-600 fs-5">
    Mobile no
  </Label>
  <Input
    id="phoneno"
    name="phoneno"
    placeholder=""
    value={obj.phoneno || ""}
    type="number"
    className="main"
    onChange={mainData}
  />
  <span className="text-danger">{iserror.phoneno}</span>
</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"
      />
    </div>
  </div>
</Col>
</Row>
```

```
        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="exampleCheck4"
        name="language"
        type="checkbox"
        className="language me-2"
        value="react"
        onChange={mainData}
    />
```

```
        checked={obj.language.includes("react")}
    />
    <Label check for="exampleCheck4" className="px-2">
        REACT
    </Label>
</div>
<span className="text-danger">{iserror.language}</span>
</Col>
<Col md={6}>
    <Label for="example" className="fw-600 fs-5">
        Gender
    </Label>
    <div className="d-flex">
        <div>
            <Input
                id="exampleCheck5"
                name="gender"
                type="radio"
                className="gender me-2"
                value="male"
                onChange={mainData}
                checked={obj.gender === "male"}>
            />
            <Label check for="exampleCheck5" 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>
    <span className="text-danger">{iserror.gender}</span>
</Col>
```

```
<Col md={12}>
  <FormGroup>
    <Label for="information" className="fw-600 fs-5">
      Information
    </Label>
    <Input
      id="information"
      name="information"
      placeholder=""
      value={obj.information || ""}
      type="textarea"
      className="main"
      onChange={mainData}
    />
    <span className="text-danger">{iserror.information}</span>
  </FormGroup>
</Col>
</Row>
<div className="text-center">
  <button className="my-2 btn btn-secondary submit fs-4">
    Submit
  </button>
</div>
</Form>
</Container>
</Col>
</Row>
<Container>
  <MainTable
    mainArray={array}
    filterArray={filterArray}
    editfun={editfun}
    deletefun={deletefun}
  />
</Container>
</Fragment>
);
};

export default MainForm;
```

Main Table.jsx

```
import React, { Fragment } from "react";
import { Table } from "reactstrap";

function MainTable(props) {
  return (
    <Fragment>
      <div className="my-5">
        <h1 className="text-center py-3">Employee Information</h1>
        <Table className="p-3">
          <thead>
            <tr>
              <th>Sr No</th>
              <th>Profile</th>
              <th>Name</th>
              <th>Last Name</th>
              <th>Email</th>
              <th>Phone No</th>
              <th>Gender</th>
              <th>Language</th>
              <th>Action</th>
            </tr>
          </thead>
          <tbody>
            {/* main array ni jagyae direct filter array par map chalavyo che so
            jo koi filter check nai karyu hoy tyare badha undefine hase ane full array a
            filter array ma return aavse ane a table ma dekhase ane jyare filter karsu tyare
            to filter array j batavano che */}
            {/* {console.log(props.filterArray)} */}
            {props.filterArray.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.lastname}</td>
                  <td>{x.email}</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;

```

validation.json

```

[
{
  "name": "username",
  "conditions": [
    {
      "condition": "!obj.username || obj.username === ''",
      "error": "User name is Required"
    },
    {
      "condition": "obj.username && !regex.test(obj.username)",
      "error": "UserName should contains only alphabet"
    }
  ]
},
{
  "name": "lastname",
  "conditions": [
    {
      "condition": "!obj.lastname || obj.lastname === ''",
      "error": "Last Name is Required"
    }
  ]
}
]
```

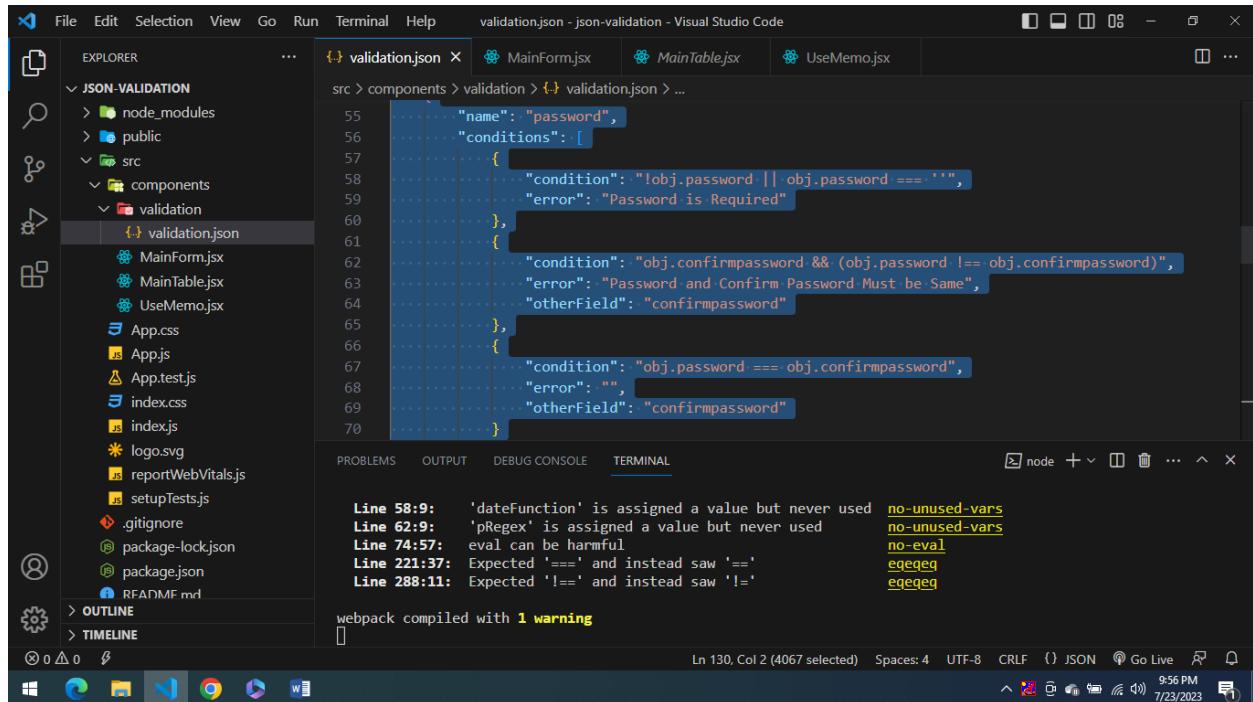
```
        "error": "Last name is Required"
    },
    {
        "condition": "obj.lastname && !regex.test(obj.lastname)",
        "error": "lastName should contains only alphabet"
    }
]
},
{
    "name": "email",
    "conditions": [
        {
            "condition": "!obj.email || obj.email === ''",
            "error": "Email is Required"
        },
        {
            "condition": "obj.email && !obj.email.includes('@gmail.com') &&
!obj.email.includes('@outlook.com') &&!obj.email.includes('@mailinator.com')",
            "error": "Email Should include @gmail or @outlook.com"
        }
    ]
},
{
    "name": "date",
    "conditions": [
        {
            "condition": "!obj.date",
            "error": "Date is Required"
        },
        {
            "condition": "!(dateFunction(new Date(obj.date).getFullYear(),
new Date(obj.date).getMonth(), new Date(obj.date).getDate()))",
            "error": "Age Should be 18+"
        }
    ]
},
{
    "name": "password",
    "conditions": [
        {
            "condition": "!obj.password || obj.password === ''",
            "error": "Password is Required"
        },
        {

```

```
        "condition": "obj.confirmpassword && (obj.password !== obj.confirmpassword)",
        "error": "Password and Confirm Password Must be Same",
        "otherField": "confirmpassword"
    },
    {
        "condition": "obj.password === obj.confirmpassword",
        "error": "",
        "otherField": "confirmpassword"
    }
]
},
{
    "name": "confirmpassword",
    "conditions": [
        {
            "condition": "!obj.confirmpassword || obj.confirmpassword === ''",
            "error": "Confirm Password is Required"
        },
        {
            "condition": "obj.password !== obj.confirmpassword",
            "error": "Password and Confirm password must be same"
        }
    ]
},
{
    "name": "phoneno",
    "conditions": [
        {
            "condition": "!obj.phoneno || obj.phoneno === ''",
            "error": "Phone Number is Required"
        },
        {
            "condition": "obj.phoneno && (obj.phoneno <= 0 || obj.phoneno.length != 10)",
            "error": "Please Enter valid Mobile No"
        }
    ]
},
{
    "name": "gender",
    "conditions": [
        {
            "condition": "!obj.gender || obj.gender === ''",
            "error": "Gender is Required"
        }
    ]
}
```

```
        "error": "Gender is Required"
    }
]
},
{
  "name": "language",
  "conditions": [
    {
      "condition": "!obj.language || obj.language.length < 3",
      "error": "atleast 3 language is Required"
    }
  ]
},
{
  "name": "information",
  "conditions": [
    {
      "condition": "!obj.information || obj.information === ''",
      "error": "information is Required"
    },
    {
      "condition": "!obj.information || obj.information === ''",
      "error": "information is Required"
    }
  ]
}
]
```

Mehru



Use Custom

CustomHook.jsx

```
import { useState } from "react"

/* how custom hook we can make */
function useA(value) {
    let [first, setfirst] = useState(value)
    function changevalue(a) {
        setfirst(a)
    }
    return [first, changevalue]
}
export default useA

/* if we want object with particular key only */
// function useCustom(value) {
//     return value.map((x) => {
//         return {
//             name:x.name,
//             age:x.age
//         }
//     })
}
```

```
//      })
// }
// export default useCustom
```

MainPage.jsx

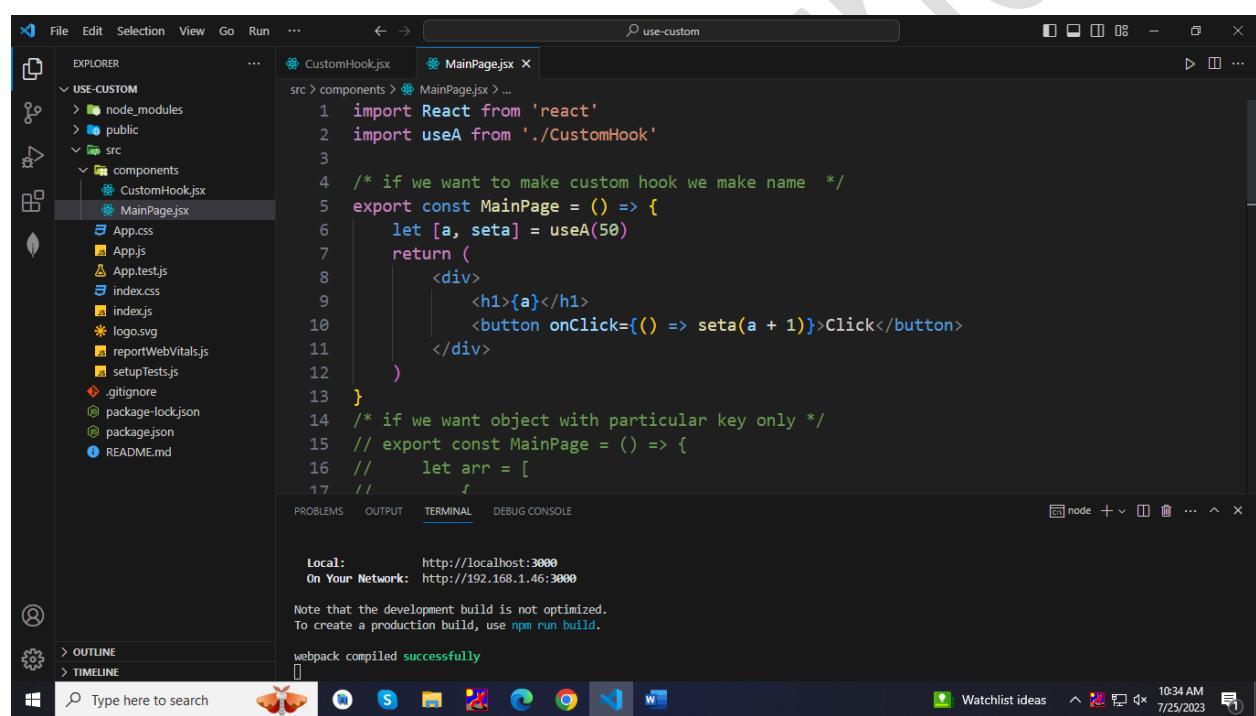
```
import React from 'react'
import useA from './CustomHook'

/* if we want to make custom hook we make name */
export const MainPage = () => {
  let [a, seta] = useA(50)
  return (
    <div>
      <h1>{a}</h1>
      <button onClick={() => seta(a + 1)}>Click</button>
    </div>
  )
}

/* if we want object with particular key only */
// export const MainPage = () => {
//   let arr = [
//     {
//       name: "mehul",
//       age: 26,
//       address: 'http'
//     },
//     {
//       name: "pratik",
//       age: 26,
//       address: 'http'
//     },
//     {
//       name: "mehul",
//       age: 26,
//       address: 'http'
//     },
//     {
//       name: "mehul",
//       age: 26,
//       form: 'http',
//     }
//   ]
// }
```

```
//           address: 'http'
//         }
//       ]
//     let a = useCustom(arr)
//     return (
//       <div>
//         <h1>{console.log(a)}</h1>
//         {/* <button onClick={() => seta(a + 1)}>Click</button> */}
//       </div>

//     )
// }
```



Memo use as a wrapper

Memo parent

```
import React, { Fragment, useCallback, useState } from 'react'
import Memo from './Memo'

const MemoParent = () => {
  const [first, setfirst] = useState(10)
  const [number, setnumber] = useState(50)
  /* it is only work when we send state as a props.if we send function as a
  props then it can not stop rerendering of child component that we can stop using
  call back */
  /*memo use in child component when we pass from parent to child and when
  parent component render but which value we pass to child as a props that is same
  then it is unnecessary render therefore to stop unnecessary rendering of child
  component we use memo wrapper in childcomponent during export */
  return (
    <Fragment>
      {console.log("Memo Parent")}
      <h1>Memo Parent</h1>
      <h1>First:{first}</h1>
      <Memo number={number} />
      <button onClick={() => setfirst(first + 1)}>Change number</button>
    </Fragment>
  )
}
export default MemoParent
```

memo child

```
import React, { memo } from 'react'

const Memo = (props) => {
  console.log("memo");
  // console.log(props.ChangeNumber);
  console.log(props.number)
  return (
    <div>
      <h2>
        Memo
    </h2>
  
```

```

        </h2>
      <h2>
        number:{props.number}
      </h2>

    </div>
  )
}

export default memo(Memo)

```

Memo output

Memo Parent

First:13

Memo

number:50

Call Back parent

```

import React, { Fragment, useCallback, useState } from 'react'
import Memo from './Memo'
import CallBack from './CallBack'

/* when we send function using props and we want to stop unnecessary rendering
then useCallback
which stop unnecessary rerendering of child component */
const CallBackparent = () => {
  const [first, setfirst] = useState(10)

```

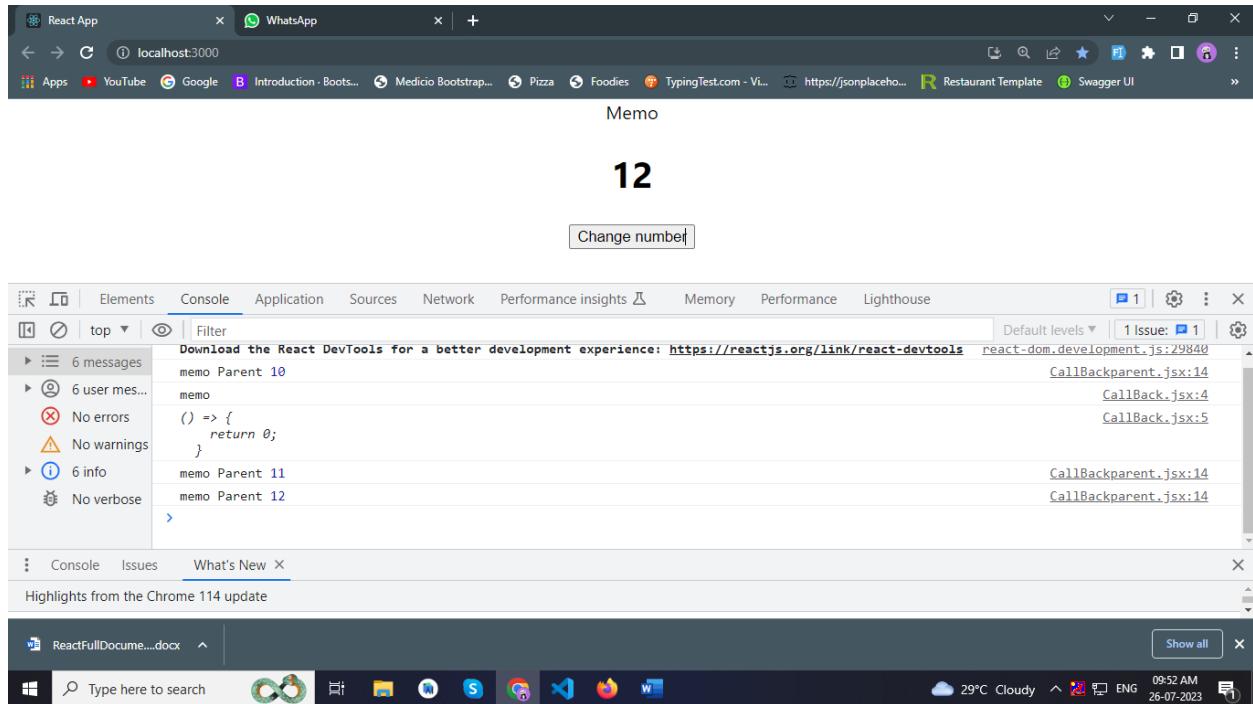
```
const ChangeNumber = useCallback(() => {
    return 0
}, [])
return (
    <Fragment>
        <CallBack ChangeNumber={ChangeNumber} />
        {console.log("memo Parent", first)}
        <h1>{first}</h1>
        <button onClick={() => setfirst(first + 1)}>Change number</button>
    </Fragment>
)
}
export default CallBackparent
```

Call back child

```
import React, { memo } from 'react'

const CallBack = (props) => {
    console.log("memo");
    console.log(props.ChangeNumber);
    return (
        <div>Memo</div>
    )
}
export default memo(CallBack)
```

Call back output



Call Back task parent

```

import React, { useCallback, useState } from 'react'
import Task from './Task'

/* in this task we pass different function on task component.that function call
from child component and which function is use from child component that
component only render other child component not render that time for that three
term is required
1=>memo use
2=>call back use
3=>dependency use */
export const TaskParent = () => {
    const [first, setfirst] = useState(10)
    const [second, setsecond] = useState(15)
    const changefirst = useCallback(() => {
        setfirst(first + 1)
    }, [first])
    const changesecond = useCallback(() => {
        setsecond(second + 1)
    }, [second])
    return (
        <div>
            <Task change={changefirst}>first</Task>
            <Task change={changesecond}>second</Task>
        </div>
    )
}

```

```
        <h1>{first}</h1>
        <h1>{second}</h1>
    </div>
)
}
```

Call back task child

```
import React, { memo } from 'react'

const Task = (props) => {
    console.log(props)
    return (
        <div>
            <button onClick={() => { props.change() }}>{props.children}</button>
        </div>
    )
}
export default memo(Task)
```

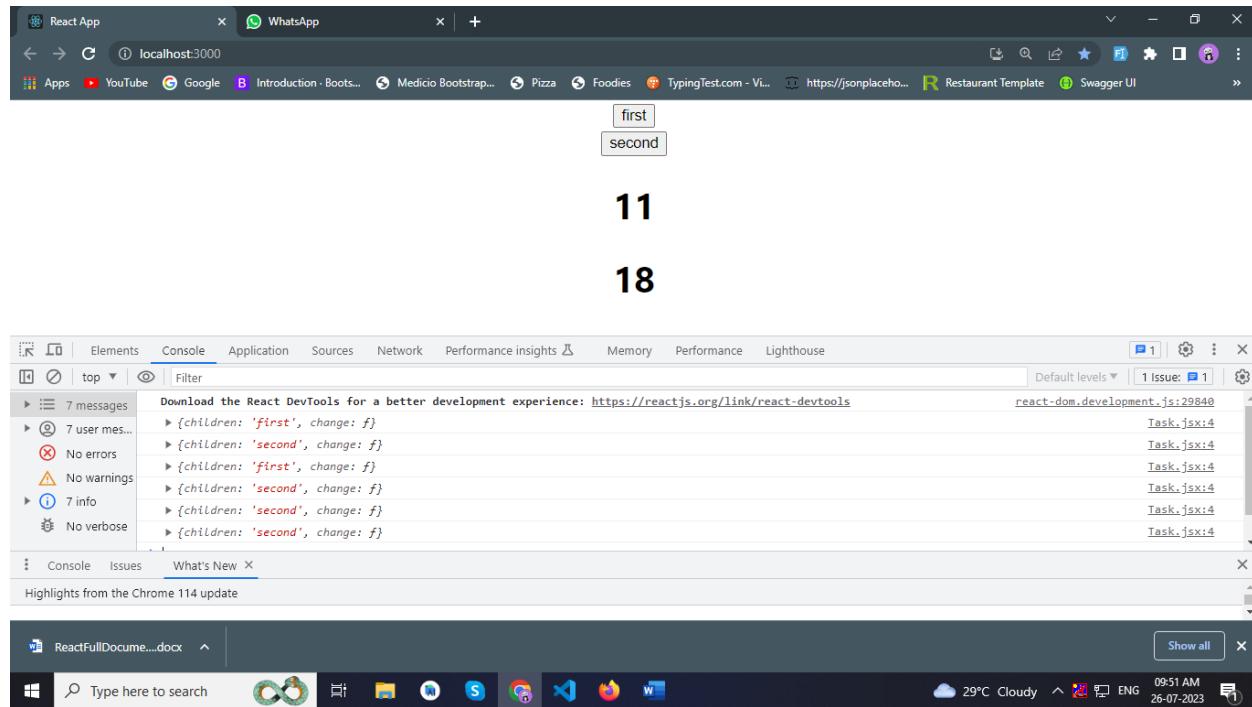
App.js

```
import logo from './logo.svg';
import './App.css';
import MemoParent from './components/MemoParent';
import CallBackparent from './components/CallBackparent';
import { TaskParent } from './components/TaskParent';

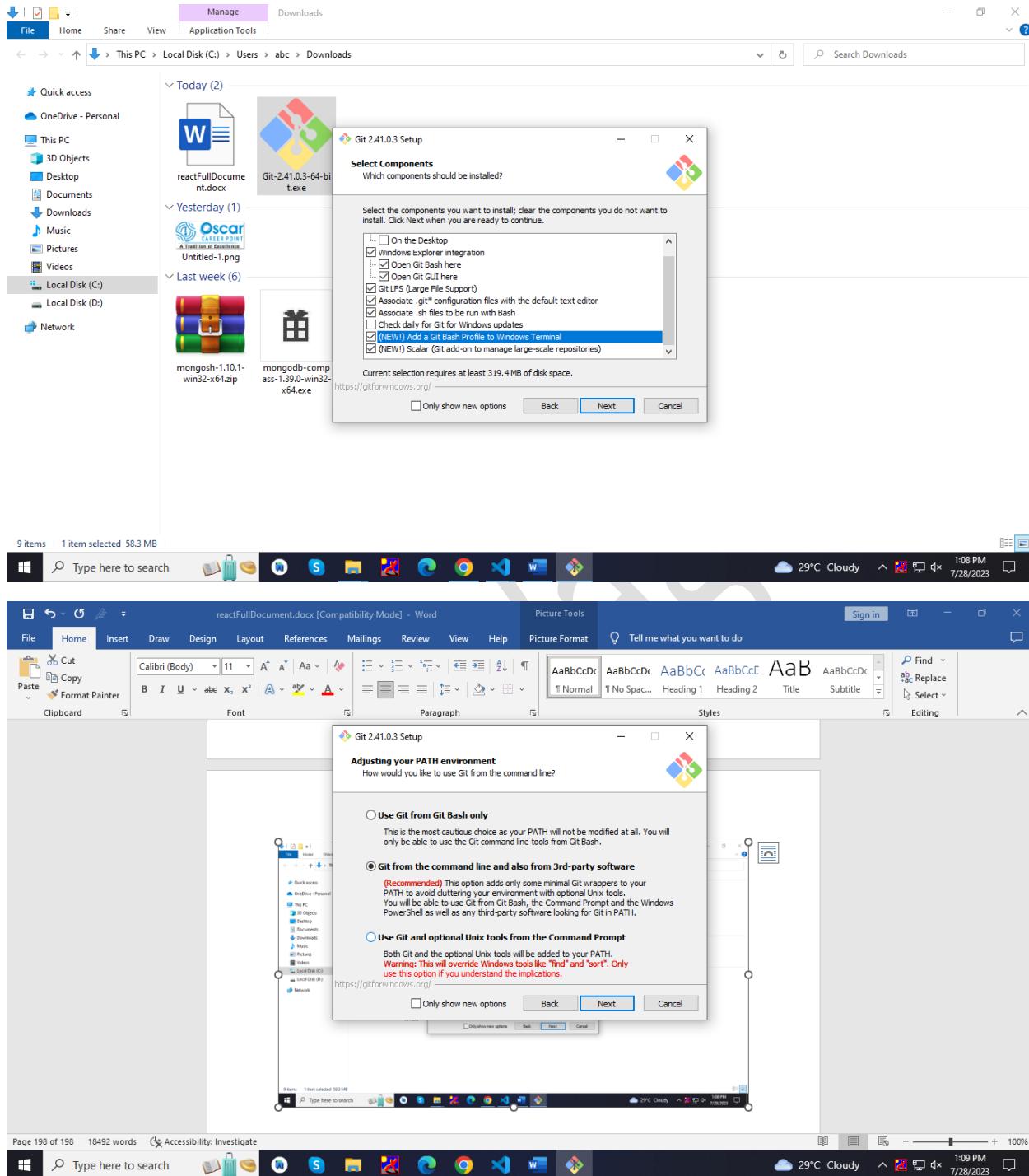
function App() {
    return (
        <div className="App">
            {/* <MemoParent /> */}
            {/* <CallBackparent /> */}
            <TaskParent />
        </div>
    );
}

export default App;
```

Output Task



Git Hub Install



```
/* to use git hub first install gitbash and when install that time set gitbash  
and other cmd can access git  
ther are two methods for set react project on github  
1=>by command
```

```
=>first open that project which we want to upload on git hub.  
=>then open terminal and check path of project and write git --version if work  
that means we can set git using vscode if not that means during install of git  
bash we had not checked on particular option that option are show in screenshot  
=>git config --global user.name "FIRST_NAME LAST_NAME"  
=>git config --global user.email "MY_NAME@example.com"  
=>sign in github  
=>then create repository and use all command given below  
  
2=>directly from vscode source control button upper side of extension */
```

Axios Api

```
import React, { Fragment, useEffect, useState } from 'react'  
import axios from 'axios'  
import { Col, Container, Form, FormGroup, Input, Label, Row, Table } from  
'reactstrap'  
const Api = () => {  
  
    Api Is stands for Application programming interface  
    For that different method and different api  
    Get method use for get data from api  
    And post method is use to set data in api  
  
    Steps to use api  
    1=>install axios by command npm I axios  
    2=>and then import it  
    3=>Free api website https://student-api.mycodelibraries.com/api-docs/#/Student/Get  
    4=>in get method first parameter is “url” of api  
    5=>.then method is run when response proper occur else catch method call  
and give error set  
    6=>then that data set in our variable  
  
    useEffect(() => {  
        axios.get('https://student-api.mycodelibraries.com/api/student/get')  
        .then((res) => {  
            arr = res.data.data  
            setarr([...arr])  
        })  
        .catch((err) => console.log(err))  
    })
```

```

        axios.post('https://student-api.mycodelibraries.com/api/student/add', obj)
        .then((res) => { console.log(res)})
        .catch((err) => console.log(err))

    }, [])

    return (
        <div>
        </div>
    )
export default Api

```

when api call when response we can show in network

The screenshot shows the Network tab in Google Chrome DevTools. A single request named "get" is listed. The request URL is <https://student-api.mycodelibraries.com/api/student/get>, the method is GET, and the status code is 200. The request took about 450ms. The Headers section includes standard CORS headers such as Access-Control-Allow-Origin set to "http://localhost:3000" and Content-Security-Policy set to "frame-ancestors 'self' * mycodelibraries.com * fizhr.com".

| Name | Headers | Preview | Response | Initiator | Timing |
|----------------|--|---------|----------|-----------|--|
| get | General Request URL: https://student-api.mycodelibraries.com/api/student/get Request Method: GET Status Code: 200 Remote Address: 116.72.16.193:443 Referrer Policy: strict-origin-when-cross-origin | | | | 50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms 450 ms 500 ms |
| 2 / 8 requests | Response Headers Access-Control-Allow-Origin: http://localhost:3000 Content-Length: 519 Content-Security-Policy: frame-ancestors 'self' * mycodelibraries.com * fizhr.com | | | | |

DevTools - localhost:3000/

Elements Console Performance insights Sources Network **Performance** Memory Application >

Preserve log Disable cache No throttling Filter Invert Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies Blocked Requests 3rd-party requests

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms 450 ms 500 ms

| Name | Headers | Preview | Response | Initiator | Timing |
|------|---------|---------|--|-----------|--------|
| get | | | ▼ { isSuccess: true, statusCode: 200, ... } ▼ data: [{firstName: "Amethyst", lastName: "Herman", hobbies: "Travelling,Reading", gender: "Female", ...}, ...] ▼ 0: {firstName: "Amethyst", lastName: "Herman", hobbies: "Travelling,Reading", gender: "Female", ...} age: 49 city: "Voluptatum illum et" createdAt: "2023-07-31T09:18:01.144Z" firstName: "Amethyst" gender: "Female" hobbies: "Travelling,Reading" lastName: "Herman" updatedAt: "2023-07-31T09:18:01.144Z" __v: 0 _id: "64c77c4901c14516c862f43c" ► 1: {firstName: "Rahil", lastName: "Diwan", hobbies: "Backpacking", gender: "male", city: "false", ...} isSuccess: true statusCode: 200 | | |
| get | | | | | |

2 / 8 requests

3:10 PM 7/31/2023

DevTools - localhost:3000/

Elements Console Performance insights Sources Network **Performance** Memory Application >

Preserve log Disable cache No throttling Filter Invert Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies Blocked Requests 3rd-party requests

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms 450 ms 500 ms

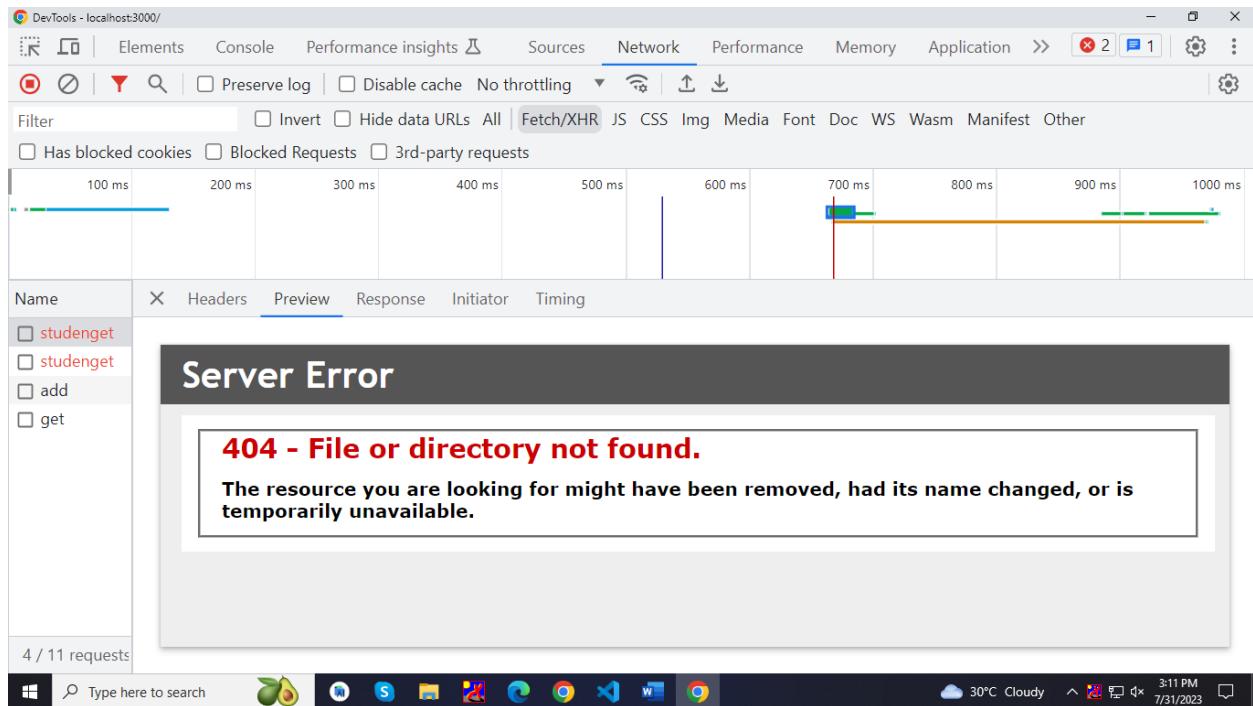
| Name | Headers | Preview | Response | Initiator | Timing |
|------|---------|---------|--|-----------|--------|
| get | | | 1 { "isSuccess": true, "statusCode": 200, "data": [{ "firstName": "Amethyst", "lastName": "Herman", "hobbies": "Travelling,Reading", "gender": "Female", "city": "Voluptatum illum et", "_id": "64c77c4901c14516c862f43c", "age": 49, "createdAt": "2023-07-31T09:18:01.144Z", "updatedAt": "2023-07-31T09:18:01.144Z", "__v": 0 }, { "firstName": "Rahil", "lastName": "Diwan", "hobbies": "Backpacking", "gender": "male", "city": "false", "_id": "64c77c4901c14516c862f43d", "age": 21, "createdAt": "2023-07-31T09:18:01.144Z", "updatedAt": "2023-07-31T09:18:01.144Z", "__v": 0 }] } | | |
| get | | | | | |

2 / 8 requests

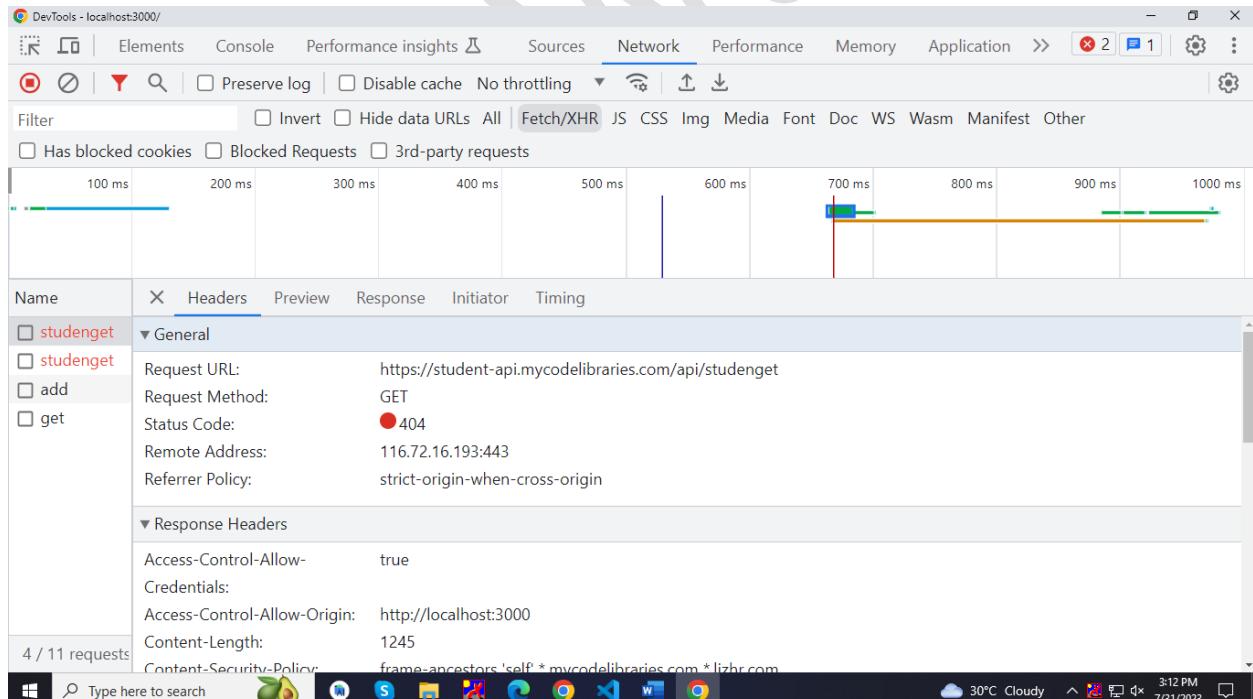
Line 1, Column 1

3:10 PM 7/31/2023

If Error Occur



The screenshot shows the Network tab in Google Chrome DevTools for a local host request. The timeline at the top indicates a total duration of approximately 1000ms. A red vertical line marks the point where the request was initiated. The request labeled "studenget" is highlighted in the list on the left. The response body shows a "Server Error" page with a red "404 - File or directory not found." message and a black text box stating, "The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable."



The screenshot shows the Network tab in Google Chrome DevTools for a remote host request. The timeline at the top indicates a total duration of approximately 1000ms. A red vertical line marks the point where the request was initiated. The request labeled "studenget" is highlighted in the list on the left. The response body shows a "Server Error" page with a red "404 - File or directory not found." message and a black text box stating, "The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable." The Headers section of the response details the following:

| Header | Value |
|-----------------------------|--|
| Request URL | https://student-api.mycodelibraries.com/api/studenget |
| Request Method | GET |
| Status Code | 404 |
| Remote Address | 116.72.16.193:443 |
| Referrer Policy | strict-origin-when-cross-origin |
| Access-Control-Allow- | true |
| Credentials | |
| Access-Control-Allow-Origin | http://localhost:3000 |
| Content-Length | 1245 |
| Content-Security-Policy | frame-ancestors 'self' * mycodelibraries.com * fizhr.com |

Form Filter

Mainform.jsx

```
import { Fragment, useEffect, useState } from "react";
import { Col, Container, Form, FormGroup, Input, Label, Row } from "reactstrap";
import MainTable from "./MainTable";
import { AiOutlineArrowDown, AiOutlineArrowUp } from "react-icons/ai";
const MainForm = () => {
  let [count, setcount] = useState(1);
  const [regex, setregex] = useState(/^[A-Za-z]+$/);
  let [obj, setobj] = useState({
    username: "",
    lastname: "",
    email: "",
    password: "",
    confirmpassword: "",
    gender: "",
    phoneno: "",
    date: "",
    information: "",
    language: [],
  });
  let [editobj, seteditobj] = useState({
    language: [],
  });
  let [array, setarray] = useState([]);
  let [iserror, setiserror] = useState({});
  let [formHeight, setformHeight] = useState(false);
  let [filterObj, setFilterObj] = useState({
    filterFname: "",
    filterLName: "",
    filterEmail: "",
  });
  let [filterArray, setfilterArray] = useState([]);

  /* to capitalize first letter of each word */
  const capitalize = (str) =>
    str
      .toLowerCase()
      .replace(/\w{1,}/g, (match) =>
        match.replace(/\w/, (m) => m.toUpperCase())
      );
}
```

```

const mainData = (e) => {
  if (e.target.name === "language") {
    if (e.target.checked) {
      obj.language.push(e.target.value);
    } else {
      obj.language = obj.language?.filter((x, i) => x !== e.target.value);
    }
  } else if (e.target.name === "information") {
    obj[e.target.name] = capitalize(e.target.value);
  } else {
    obj[e.target.name] = e.target.value;
  }

  setobj({ ...obj });
  validate(e.target.name);
};

const dateFunction = (year, month, day) => {
  return new Date(year + 18, month, day) <= new Date();
};
const validate = (name) => {
  /* ===== For User Name===== */
  if (name == "username") {
    if (!obj.username || obj.username === "") {
      iserror.username = "FirstName Should Not be empty";
    } else if (obj.username && !regex.test(obj.username)) {
      iserror.username = "FirstName should contains only alphabet";
    } else {
      delete iserror.username;
    }
  }
  /* ===== For User Name===== */
  if (name == "lastname") {
    if (!obj.lastname || obj.lastname === "") {
      iserror.lastname = "LastName Should Not be empty";
    } else if (obj.lastname && !regex.test(obj.lastname)) {
      iserror.lastname = "lastName should contains only alphabet";
    } else {
      delete iserror.lastname;
    }
  }
  /* ===== For Email===== */
  if (name == "email") {
    if (!obj.email || obj.email === "") {

```

```

iserror.email = "Email Should Not be empty";
} else if (
    obj.email &&
    !obj.email.includes("@gmail.com") &&
    !obj.email.includes("@outlook.com") &&
    !obj.email.includes("@mailinator.com")
) {
    iserror.email = "Email Should include @gmail or @outlook.com";
} else {
    delete iserror.email;
}
}
/* ===== For Date ===== */
if (name == "date") {
    let date = new Date(obj.date);
    if (!obj.date || obj.date === "") {
        iserror.date = "date Should Not be empty";
    } else if (
        !dateFunction(date.getFullYear(), date.getMonth(), date.getDate())
    ) {
        iserror.date = "Age Should be 18+";
    } else {
        delete iserror.date;
    }
}

// else if (
//     obj.date &&
//     (new Date().getTime() - new Date(obj.date).getTime()) /
//     (1000 * 3600 * 24 * 365) <=
//     18
// ) {
//     iserror.date = "Age Should be 18+";
// }

/* ===== For Password ===== */
/*===== For Confirm Password ===== */
if (name == "password" || name == "confirmpassword") {
    if (!obj.password || obj.password === "") {
        iserror.password = "Password Should Not Be Empty";
    } else if (
        obj.password &&
        !/^([a-z]+[A-Z]+[0-9]+[$!%*?&]{8,})$/.test(
            obj.password
        )
    ) {
        iserror.password = "Password Should Not Be Empty";
    }
}

```

```

        )
    ) {
        iserror.password =
            "Minimum eight,at least one uppercase,lowercase letter,one number and
one special character:";
    } else {
        delete iserror.password;
    }
    if (!obj.confirmpassword || obj.confirmpassword === "") {
        iserror.confirmpassword = "Confirm Password Should Not Be Empty";
    } else if (!(obj.confirmpassword === obj.password)) {
        iserror.confirmpassword =
            "Password and Confirm Password Should Be Same";
    } else {
        delete iserror.confirmpassword;
    }
}

/*===== For Mobile Number =====*/
if (name == "phoneno") {
    if (!obj.phoneno || obj.phoneno === "") {
        iserror.phoneno = "Mobile number should not be empty";
    } else if (
        obj.phoneno &&
        (obj.phoneno <= 0 || obj.phoneno.length != 10)
    ) {
        iserror.phoneno = "Please Enter valid Mobile No";
    } else {
        delete iserror.phoneno;
    }
}
/*===== For Gender =====*/
if (name == "gender") {
    if (!obj.gender) {
        iserror.gender = "Gender is Required";
    } else {
        delete iserror.gender;
    }
}

/*===== For Language =====*/
if (name == "language") {
    if (!obj.language || obj.language.length < 3) {
        iserror.language = "Please Select atleast Three";
    } else {

```

```

        delete iserror.language;
    }
}

/*===== For Information ===== */
if (name == "information") {
    if (!obj.information || obj.information === "") {
        iserror.information = "Information is required";
    } else {
        delete iserror.information;
    }
}
setiserror({ ...iserror });
};

const submitFunction = (e) => {
    e.preventDefault();
    /* to send all key for validation when form is blank and we click submit */
    for (let key in obj) {
        validate(key);
    }
    /* if we fill all field true in form all object key is delete and iserror
object become blank. to check blank we convert array key in to array as below. we
can not compare blank object with blank object*/
    if (Object.keys(iserror).length == 0) {
        if (obj.id === undefined) {
            count++;
            obj["id"] = count;
            array = [...array, obj];

            setcount(count);
        } else {
            array.splice(
                array.findIndex((x) => x.id === obj.id),
                1,
                obj
            );
        }
    }
    localStorage.setItem("array", JSON.stringify(array));
    localStorage.setItem("count", count);
    obj = {
        username: "",
        lastname: "",
        email: "",
        phoneno: ""
    };
}

```

```
    pincode: "",
    gender: "",
    language: [],
};

setobj({ ...obj });
setarray([...array]);
filterArray = array;
setfilterArray([...filterArray]);
filterFunction(1);
};

function editfun(id) {
    editobj = array.find((x) => x.id === id);
    setobj({ ...editobj });
}
function deletefun(id) {
    array.splice(
        array.findIndex((x) => x.id === id),
        1
    );
    filterArray = array;
    setarray([...array]);
    setfilterArray([...filterArray]);

    localStorage.setItem("array", JSON.stringify(array));
    filterFunction(1);
}

useEffect(() => {
    setarray(JSON.parse(localStorage.getItem("array")) || []);
    setfilterArray(JSON.parse(localStorage.getItem("array")) || []);
    setcount(JSON.parse(localStorage.getItem("count")) || 0);
}, []);

const formOpen = () => {
    formHeight = true;
    setformHeight(formHeight);
};

/* when we want to form close we set height in which only form heading show and
style overflow hidden */
const formClose = () => {
    formHeight = false;
    setformHeight(formHeight);
};

const filterFunction = (e) => {
```

```

/* jyare delete edit ke submit kariye tyare pan filter thatu rakhva mate
filterfunction call karyu che submit na function par pan tyare aapri pase koi key
hoti nathi ane aapne key ni jarur nathi pan key nai male to e.target.name ma
error aape atle aapre tya one pass karyo che so jo e ma one aave to upar ni 2
line nu work stop thai jase */
if (e != 1) {
    filterObj[e.target.name] = e.target.value;
    setFilterObj({ ...filterObj });
}
filterArray = array.filter(
    (x) =>
        /* if x.lastname or any other data is not available and we give &&
condition between them then it is not check that undefined data condition or skip
it therefore if any one field of filter we fill then only that condition check
other is skip and filter only according to one condition. when filter by two data
then two condition and another one skip nad so on */
        x.username
            .toLowerCase()
            .includes(filterObj.filterFname.toLowerCase()) &&
        x.lastname
            .toLowerCase()
            .includes(filterObj.filterLName.toLowerCase()) &&
        x.email.toLowerCase().includes(filterObj.filterEmail.toLowerCase())
);

// filterArray = array.filter((x) => {
//     if (filterObj.filterFname?.length != 0 || filterObj.filterLName?.length
!= 0 || filterObj.filterEmail?.length != 0) {
//         if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0 && filterObj.filterEmail?.length != 0) {
//             if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
//                 return x;
//             }
//         }
//         else if (filterObj.filterLName?.length != 0 &&
filterObj.filterEmail?.length != 0) {
//             if (x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
//                 return x;
//             }
//         }
//         else if (filterObj.filterFname?.length != 0 &&
filterObj.filterEmail?.length != 0) {

```

```

        //      if (x.username.includes(filterObj.filterFname) &&
x.email.includes(filterObj.filterEmail)) {
        //          return x;
        //      }
        //      }
        //      else if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0) {
        //          if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName)) {
        //              return x;
        //          }
        //      }
        //      else if (filterObj.filterLName?.length != 0) {
        //          if (x.lastname.includes(filterObj.filterLName)) {
        //              return x;
        //          }
        //      }
        //      else if (filterObj.filterEmail?.length != 0) {
        //          if (x.email.includes(filterObj.filterEmail)) {
        //              return x;
        //          }
        //      }
        //      else if (filterObj.filterFname?.length != 0) {
        //          if (x.username.includes(filterObj.filterFname)) {
        //              return x;
        //          }
        //      }
        //  }
        //  else {
        //      return x
        //  }
    // })
    setfilterArray(filterArray);
};

return (
<Fragment>
<div className="container mt-2 mb-5 border border-1 border-black rounded p-2">
    <h1 className="text-center"> Form Filter </h1>
    <div className="row">
        <div className="col-4">
            <FormGroup>
                <Label htmlFor="filterFname" className="fw-600 fs-5">
                    User Name
                </Label>

```

```
<Input
  id="filterFname"
  name="filterFname"
  placeholder="User name"
  type="text"
  onChange={filterFunction}
/>
<span className="text-danger">{iserror.email}</span>
</FormGroup>
</div>
<div className="col-4">
<FormGroup>
<Label for="lastName" className="fw-600 fs-5">
  Last Name
</Label>
<Input
  id="lastName"
  name="filterLName"
  placeholder="Last Name"
  type="text"
  className="main"
  onChange={filterFunction}
/>
<span className="text-danger">{iserror.email}</span>
</FormGroup>
</div>
<div className="col-4">
<FormGroup>
<Label for="filterEmail" className="fw-600 fs-5">
  Email
</Label>
<Input
  id="filterEmail"
  name="filterEmail"
  placeholder="Filter email"
  type="email"
  onChange={filterFunction}
/>
<span className="text-danger">{iserror.email}</span>
</FormGroup>
</div>
</div>
<Row>
<Col xs={6} className="offset-3">
```

```
<Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
  <h1 className="text-center py-3">
    Employee Form{" "}
    {formHeight ? (
      <AiOutlineArrowUp onClick={formClose} />
    ) : (
      <AiOutlineArrowDown onClick={formOpen} />
    )}
  </h1>
  <Form
    onSubmit={(e) => {
      submitFunction(e);
    }}
    style={{
      height: formHeight ? "auto" : "0px",
      overflow: "hidden",
    }}
  >
    <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}
          />
          <span className="text-danger">{iserror.username}</span>
        </FormGroup>
      </Col>
      <Col md={6}>
        <FormGroup>
          <Label for="lastname" className="fw-600 fs-5">
            Last Name
          </Label>
          <Input
            id="lastname"
            name="lastname"
          />
        </FormGroup>
      </Col>
    </Row>
  </Form>
</Container>
```

```
        placeholder=""
        type="text"
        className="main"
        value={obj.lastname || ""}
        onChange={mainData}
    />
    <span className="text-danger">{iserror.username}</span>
</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}
    />
    <span className="text-danger">{iserror.email}</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="date" className="fw-600 fs-5">
        Birth Date
    </Label>
    <Input
        id="date"
        name="date"
        placeholder=""
        type="date"
        value={obj.date || ""}
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">{iserror.date}</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
```

```
<Label for="password" className="fw-600 fs-5">
    Password
</Label>
<Input
    id="password"
    name="password"
    placeholder=""
    type="password"
    value={obj.password || ""}
    className="main"
    onChange={mainData}
/>
<span className="text-danger">{iserror.password}</span>
</FormGroup>
</Col>
<Col md={6}>
    <FormGroup>
        <Label for="confirmpassword" className="fw-600 fs-5">
            Confirm Password
        </Label>
        <Input
            id="confirmpassword"
            name="confirmpassword"
            placeholder=""
            type="password"
            value={obj.confirmpassword || ""}
            className="main"
            onChange={mainData}
        />
        <span className="text-danger">
            {iserror.confirmpassword}
        </span>
    </FormGroup>
</Col>
<Col md={6}>
    <FormGroup>
        <Label for="phoneno" className="fw-600 fs-5">
            Mobile no
        </Label>
        <Input
            id="phoneno"
            name="phoneno"
            placeholder=""
            value={obj.phoneno || ""}
            type="number"
        />
    </FormGroup>
</Col>
```

```
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">{iserror.phoneno}</span>
</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>
</Row>
```

```
</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="exampleCheck4"
    name="language"
    type="checkbox"
    className="language me-2"
    value="react"
    onChange={mainData}
    checked={obj.language.includes("react")}
  />
  <Label check for="exampleCheck4" className="px-2">
    REACT
  </Label>
</div>
<span className="text-danger">{iserror.language}</span>
</Col>
<Col md={6}>
  <Label for="example" className="fw-600 fs-5">
    Gender
  </Label>
  <div className="d-flex">
    <div>
      <Input
        id="exampleCheck5"
        name="gender"
        type="radio"
        className="gender me-2"
        value="male"
        onChange={mainData}
      />
    </div>
    <div>
      <Input
        id="exampleCheck6"
        name="gender"
        type="radio"
        className="gender me-2"
        value="female"
        onChange={mainData}
      />
    </div>
  </div>
</Col>
```

```
        checked={obj.gender === "male"}  
    />  
    <Label check for="exampleCheck5" 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>  
  <span className="text-danger">{iserror.gender}</span>  
</Col>  
<Col md={12}>  
  <FormGroup>  
    <Label for="information" className="fw-600 fs-5">  
        Information  
    </Label>  
    <Input  
        id="information"  
        name="information"  
        placeholder=""  
        value={obj.information || ""}  
        type="textarea"  
        className="main"  
        onChange={mainData}  
    />  
    <span className="text-danger">{iserror.information}</span>  
  </FormGroup>  
</Col>  
</Row>  
<div className="text-center">  
  <button className="my-2 btn btn-secondary submit fs-4">  
      Submit  
  </button>
```

```

        </div>
      </Form>
    </Container>
  </Col>
</Row>
<Container>
  <MainTable
    mainArray={array}
    filterArray={filterArray}
    editfun={editfun}
    deletefun={deletefun}
  />
</Container>
<Fragment>
);
};

export default MainForm;

```

Main Table.jsx

```

import React, { Fragment } from "react";
import { Table } from "reactstrap";

function MainTable(props) {
  return (
    <Fragment>
      <div className="my-5">
        <h1 className="text-center py-3">Employee Information</h1>
        <Table className="p-3">
          <thead>
            <tr>
              <th>Sr No</th>
              <th>Profile</th>
              <th>Name</th>
              <th>LastName</th>
              <th>Email</th>
              <th>Phone No</th>
              <th>Gender</th>
              <th>Language</th>
              <th>Action</th>
            </tr>
          </thead>
          <tbody>
            /* main array ni jagyae direct filter array par map chalavyo che so
           jo koi filter check nai karyu hoy tyare badha undefined hase ane full array a

```

```

filter array ma return aavse ane a table ma dekhase ane jyare filter karsu tyare
to filter array j batavano che */
    /* {console.log(props.filterArray)} */
{props.filterArray.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.lastname}</td>
            <td>{x.email}</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;
```

Form Validation Json

Form.jsx

```
[

  {

    "name": "username",
    "conditions": [
      {
        "condition": "!obj.username || obj.username === ''",
        "error": "User name is Required"
      },
      {
        "condition": "obj.username && !regex.test(obj.username)",
        "error": "UserName should contains only alphabet"
      }
    ]
  },
  {
    "name": "lastname",
    "conditions": [
      {
        "condition": "!obj.lastname || obj.lastname === ''",
        "error": "Last name is Required"
      },
      {
        "condition": "obj.lastname && !regex.test(obj.lastname)",
        "error": "lastName should contains only alphabet"
      }
    ]
  },
]
```

```
{
    "name": "email",
    "conditions": [
        {
            "condition": "!obj.email || obj.email === ''",
            "error": "Email is Required"
        },
        {
            "condition": "obj.email && !obj.email.includes('@gmail.com') &&
!obj.email.includes('@outlook.com') &&!obj.email.includes('@mailinator.com')",
            "error": "Email Should include @gmail or @outlook.com"
        }
    ]
},
{
    "name": "date",
    "conditions": [
        {
            "condition": "!obj.date",
            "error": "Date is Required"
        },
        {
            "condition": "!(dateFunction(new Date(obj.date).getFullYear(),
new Date(obj.date).getMonth(), new Date(obj.date).getDate()))",
            "error": "Age Should be 18+"
        }
    ]
},
{
    "name": "password",
    "conditions": [
        {
            "condition": "!obj.password || obj.password === ''",
            "error": "Password is Required"
        },
        {
            "condition": "obj.confirmPassword && (obj.password !==
obj.confirmPassword)",
            "error": "Password and Confirm Password Must be Same",
            "otherField": "confirmPassword"
        },
        {
            "condition": "obj.password === obj.confirmPassword",
            "error": "",
            "otherField": "confirmPassword"
        }
    ]
}
```

```
        }
    ],
},
{
    "name": "confirmPassword",
    "conditions": [
        {
            "condition": "!obj.confirmPassword || obj.confirmPassword === '',
            "error": "Confirm Password is Required"
        },
        {
            "condition": "obj.password !== obj.confirmPassword",
            "error": "Password and Confirm password must be same"
        }
    ]
},
{
    "name": "phoneno",
    "conditions": [
        {
            "condition": "!obj.phoneno || obj.phoneno === '',
            "error": "Phone Number is Required"
        },
        {
            "condition": "obj.phoneno && (obj.phoneno <= 0 ||
obj.phoneno.length != 10)",
            "error": "Please Enter valid Mobile No"
        }
    ]
},
{
    "name": "gender",
    "conditions": [
        {
            "condition": "!obj.gender || obj.gender === '',
            "error": "Gender is Required"
        }
    ]
},
{
    "name": "language",
    "conditions": [
        {
            "condition": "!obj.language || obj.language.length < 3",
            "error": "Language is Required"
        }
    ]
}
```

```

        "error": "atleast 3 language is Required"
    }
]
},
{
    "name": "information",
    "conditions": [
        {
            "condition": "!obj.information || obj.information === ''",
            "error": "information is Required"
        },
        {
            "condition": "!obj.information || obj.information === ''",
            "error": "information is Required"
        }
    ]
}
]
```

Mainform.jsx

```

import { Fragment, useEffect, useState } from "react";
import { Col, Container, Form, FormGroup, Input, Label, Row } from "reactstrap";
import MainTable from "./MainTable";
import { AiOutlineArrowDown, AiOutlineArrowUp } from "react-icons/ai";
import ValidationData from '../components/validation/validation.json'
const MainForm = () => {
    let [count, setcount] = useState(1);
    const [regex, setregex] = useState(/^[A-Za-z]+$/);
    let [obj, setobj] = useState({
        username: "",
        lastname: "",
        email: "",
        password: "",
        confirmpassword: "",
        gender: "",
        phoneno: "",
        date: "",
        information: "",
        language: []
    });
    let [editobj, seteditobj] = useState({
        language: []
    });
}
```

```

let [array, setarray] = useState([]);
let [iserror, setIserror] = useState({});
let [formHeight, setformHeight] = useState(false);
let [filterObj, setFilterObj] = useState({
  filterFname: "",
  filterLName: "",
  filterEmail: ""
});
let [filterArray, setfilterArray] = useState([]);
/* to capitalize first letter of each word */
const capitalize = (str) =>
  str
    .toLowerCase()
    .replace(/\w{1,}/g, (match) =>
      match.replace(/\w/, (m) => m.toUpperCase())
    );
const mainData = (e) => {
  if (e.target.name === "language") {
    if (e.target.checked) {
      obj.language.push(e.target.value);
    } else {
      obj.language = obj.language?.filter((x, i) => x !== e.target.value);
    }
  } else if (e.target.name === "information") {
    obj[e.target.name] = capitalize(e.target.value);
  } else {
    obj[e.target.name] = e.target.value;
  }
  setobj({ ...obj });
  validate(e.target.name);
};

const dateFunction = (year, month, day) => {
  return new Date(year + 18, month, day) <= new Date();
};
const validate = (name) => {
  let pRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$!%*?&])[A-Za-zA\d$!%*?&]{8,}$/;
  /* if we write iserror.name= 'djfshd' then it is take name as a property not
take as a variable
  and if we write iserror[name] then it takes name as variable and make
property according variable value

```

```
why we use find method in validation obj?  
because find method find only one element of array and therefore when change  
event call that time in name only that field name occur and find method find only  
that obj.
```

```
why we use find method in errorobj?  
because if we use for each method then it is check all the condition and we  
want if and else if type process
```

```
if we use find then if condition become true then it is not check any next  
conditions. But for that reason we have to set erroor condition according to  
order wise like first if then else if condition and so on*/
```

```
let validationObj = ValidationData.find(x => x.name === name)  
console.log(validationObj)  
let errorObj = validationObj.conditions.find((x) => eval(x.condition))  
console.log(errorObj)  
if (validationObj) {  
    if (errorObj) {  
        if (errorObj.otherField) {  
            iserror[errorObj.otherField] = errorObj.error  
        }  
        else {  
            iserror[name] = errorObj.error  
        }  
    }  
    else {  
        delete iserror[name]  
    }  
}  
// /* ===== For User Name===== */  
// if (name == "username") {  
//     if (!obj.username || obj.username === "") {  
//         iserror.username = "FirstName Should Not be empty";  
//     } else if (obj.username && !regex.test(obj.username)) {  
//         iserror.username = "FirstName should contains only alphabet";  
//     } else {  
//         delete iserror.username;  
//     }  
// }  
// /* ===== For User Name===== */  
// if (name == "lastname") {  
//     if (!obj.lastname || obj.lastname === "") {  
//         iserror.lastname = "LastName Should Not be empty";  
//     } else if (obj.lastname && !regex.test(obj.lastname)) {  
//         iserror.lastname = "lastName should contains only alphabet";  
//     } else {
```

```
//      delete iserror.lastname;
//    }
//  }
// /* ===== For Email ===== */
// if (name == "email") {
//   if (!obj.email || obj.email === "") {
//     iserror.email = "Email Should Not be empty";
//   } else if (
//     obj.email &&
//     !obj.email.includes("@gmail.com") &&
//     !obj.email.includes("@outlook.com") &&
//     !obj.email.includes("@mailinator.com")
//   ) {
//     iserror.email = "Email Should include @gmail or @outlook.com";
//   } else {
//     delete iserror.email;
//   }
// }
// /* ===== For Date ===== */
// if (name == "date") {
//   let date = new Date(obj.date);
//   if (!obj.date || obj.date === "") {
//     iserror.date = "date Should Not be empty";
//   } else if (
//     !dateFunction(date.getFullYear(), date.getMonth(), date.getDate())
//   ) {
//     iserror.date = "Age Should be 18+";
//   } else {
//     delete iserror.date;
//   }
// }

// // else if (
// //   obj.date &&
// //   (new Date().getTime() - new Date(obj.date).getTime()) /
// //   (1000 * 3600 * 24 * 365) <=
// //   18
// // ) {
// //   iserror.date = "Age Should be 18+";
// // }

// /* ===== For Password ===== */
// /*===== For Confirm Password =====*/
// if (name == "password" || name == "confirmpassword") {
//   if (!obj.password || obj.password === "") {
```

```

//           iserror.password = "Password Should Not Be Empty";
//       } else if (
//           obj.password &&
//           !/^([a-z])([A-Z])(\d)([@$!%*?&])[A-Za-
z\d@$!%*?&]{8,}$.test(
//               obj.password
//           )
//       ) {
//           iserror.password =
//               "Minimum eight,at least one uppercase,lowercase letter,one number
and one special character:";
//       } else {
//           delete iserror.password;
//       }
//       if (!obj.confirmpassword || obj.confirmpassword === "") {
//           iserror.confirmpassword = "Confirm Password Should Not Be Empty";
//       } else if (!(obj.confirmpassword === obj.password)) {
//           iserror.confirmpassword =
//               "Password and Confirm Password Should Be Same";
//       } else {
//           delete iserror.confirmpassword;
//       }
//   }

//   /*===== For Mobile Number ===== */
//   if (name == "phoneno") {
//       if (!obj.phoneno || obj.phoneno === "") {
//           iserror.phoneno = "Mobile number should not be empty";
//       } else if (
//           obj.phoneno &&
//           (obj.phoneno <= 0 || obj.phoneno.length != 10)
//       ) {
//           iserror.phoneno = "Please Enter valid Mobile No";
//       } else {
//           delete iserror.phoneno;
//       }
//   }
//   /*===== For Gender ===== */
//   if (name == "gender") {
//       if (!obj.gender) {
//           iserror.gender = "Gender is Required";
//       } else {
//           delete iserror.gender;
//       }
//   }

```

```

// /*===== For Language =====*/
// if (name == "language") {
//   if (!obj.language || obj.language.length < 3) {
//     iserror.language = "Please Select atleast Three";
//   } else {
//     delete iserror.language;
//   }
// }
// /*===== For Information =====*/
// if (name == "information") {
//   if (!obj.information || obj.information === "") {
//     iserror.information = "Information is required";
//   } else {
//     delete iserror.information;
//   }
// }
setiserror({ ...iserror });
};

const submitFunction = (e) => {
  e.preventDefault();
  /* to send all key for validation when form is blank and we click submit */
  for (let key in obj) {
    validate(key);
  }
  /* if we fill all field true in form all object key is delete and iserror
  object become blank. to check blank we convert array key in to array as below. we
  can not compare blank object with blank object*/
  if (Object.keys(iserror).length == 0) {
    if (obj.id === undefined) {
      count++;
      obj["id"] = count;
      array = [...array, obj];

      setcount(count);
    } else {
      array.splice(
        array.findIndex((x) => x.id === obj.id),
        1,
        obj
      );
    }
  }
  localStorage.setItem("array", JSON.stringify(array));
}

```

```

localStorage.setItem("count", count);
obj = {
  username: "",
  lastname: "",
  email: "",
  phoneno: "",
  pincode: "",
  gender: "",
  language: [],
};
setobj({ ...obj });
setarray([...array]);
filterArray = array;
setfilterArray([...filterArray]);
filterFunction(1);
};

function editfun(id) {
  editobj = array.find((x) => x.id === id);
  setobj({ ...editobj });
}
function deletefun(id) {
  array.splice(
    array.findIndex((x) => x.id === id),
    1
  );
  filterArray = array;
  setarray([...array]);
  setfilterArray([...filterArray]);

  localStorage.setItem("array", JSON.stringify(array));
  filterFunction(1);
}

useEffect(() => {
  setarray(JSON.parse(localStorage.getItem("array")) || []);
  setfilterArray(JSON.parse(localStorage.getItem("array")) || []);
  setcount(JSON.parse(localStorage.getItem("count")) || 0);
}, []);
const formOpen = () => {
  formHeight = true;
  setformHeight(formHeight);
};
/* when we want to form close we set height in which only form heading show and
style overflow hidden */

```

```

const formClose = () => {
  formHeight = false;
  setformHeight(formHeight);
};

const filterFunction = (e) => {
  /* jyare delete edit ke submit kariye tyare pan filter thatu rakhva mate
filterfunction call karyu che submit na function par pan tyare aapri pase koi key
hoti nathi ane aapne key ni jarur nathi pan key nai male to e.target.name ma
error aape atle aapre tya one pass karyo che so jo e ma one aave to upar ni 2
line nu work stop thai jase */
  if (e != 1) {
    filterObj[e.target.name] = e.target.value;
    setFilterObj({ ...filterObj });
  }
  filterArray = array.filter(
    (x) =>
      /* if x.lastname or any other data is not available and we give &&
condition between them then it is not check that undefined data condition or skip
it therefore if any one field of filter we fill then only that condition check
other is skip and filter only according to one condition. when filter by two data
then two condition and another one skip nad so on */
      x.username
        .toLowerCase()
        .includes(filterObj.filterFname.toLowerCase()) &&
      x.lastname
        .toLowerCase()
        .includes(filterObj.filterLName.toLowerCase()) &&
      x.email.toLowerCase().includes(filterObj.filterEmail.toLowerCase())
  );
}

// filterArray = array.filter((x) => {
//   if (filterObj.filterFname?.length != 0 || filterObj.filterLName?.length
!= 0 || filterObj.filterEmail?.length != 0) {
//     if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0 && filterObj.filterEmail?.length != 0) {
//       if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
//         return x;
//       }
//     }
//     else if (filterObj.filterLName?.length != 0 &&
filterObj.filterEmail?.length != 0) {

```

```
//      if (x.lastname.includes(filterObj.filterLName) &&
x.email.includes(filterObj.filterEmail)) {
//          return x;
//      }
//      }
//      else if (filterObj.filterFname?.length != 0 &&
filterObj.filterEmail?.length != 0) {
//          if (x.username.includes(filterObj.filterFname) &&
x.email.includes(filterObj.filterEmail)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterFname?.length != 0 &&
filterObj.filterLName?.length != 0) {
//          if (x.username.includes(filterObj.filterFname) &&
x.lastname.includes(filterObj.filterLName)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterLName?.length != 0) {
//          if (x.lastname.includes(filterObj.filterLName)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterEmail?.length != 0) {
//          if (x.email.includes(filterObj.filterEmail)) {
//              return x;
//          }
//      }
//      else if (filterObj.filterFname?.length != 0) {
//          if (x.username.includes(filterObj.filterFname)) {
//              return x;
//          }
//      }
//  }
//  else {
//      return x
//  }
// })
setfilterArray(filterArray);
};

return (
<Fragment>
<div className="container mt-2 mb-5 border border-1 border-black rounded p-2">
```

```
<h1 className="text-center"> Form Filter </h1>
<div className="row">
  <div className="col-4">
    <FormGroup>
      <Label for="filterFname" className="fw-600 fs-5">
        User Name
      </Label>
      <Input
        id="filterFname"
        name="filterFname"
        placeholder="User name"
        type="text"
        onChange={filterFunction}
      />
      <span className="text-danger">{iserror.email}</span>
    </FormGroup>
  </div>
  <div className="col-4">
    <FormGroup>
      <Label for="lastName" className="fw-600 fs-5">
        Last Name
      </Label>
      <Input
        id="lastName"
        name="filterLName"
        placeholder="Last Name"
        type="text"
        className="main"
        onChange={filterFunction}
      />
      <span className="text-danger">{iserror.email}</span>
    </FormGroup>
  </div>
  <div className="col-4">
    <FormGroup>
      <Label for="filterEmail" className="fw-600 fs-5">
        Email
      </Label>
      <Input
        id="filterEmail"
        name="filterEmail"
        placeholder="Filter email"
        type="email"
        onChange={filterFunction}
      />
```

```
        <span className="text-danger">{iserror.email}</span>
      </FormGroup>
    </div>
  </div>
</div>
<Row>
  <Col xs={6} className="offset-3">
    <Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
      <h1 className="text-center py-3">
        Employee Form{" "}
        {formHeight ? (
          <AiOutlineArrowUp onClick={formClose} />
        ) : (
          <AiOutlineArrowDown onClick={formOpen} />
        )}
      </h1>
      <Form
        onSubmit={(e) => {
          submitFunction(e);
        }}
        style={{
          height: formHeight ? "auto" : "0px",
          overflow: "hidden",
        }}
      >
        <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}
              />
              <span className="text-danger">{iserror.username}</span>
            </FormGroup>
          </Col>
          <Col md={6}>
```

```
<FormGroup>
  <Label for="lastname" className="fw-600 fs-5">
    Last Name
  </Label>
  <Input
    id="lastname"
    name="lastname"
    placeholder=""
    type="text"
    className="main"
    value={obj.lastname || ""}
    onChange={mainData}
  />
  <span className="text-danger">{iserror.lastname}</span>
</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}
    />
    <span className="text-danger">{iserror.email}</span>
  </FormGroup>
</Col>
<Col md={6}>
  <FormGroup>
    <Label for="date" className="fw-600 fs-5">
      Birth Date
    </Label>
    <Input
      id="date"
      name="date"
      placeholder=""
      type="date"
      value={obj.date || ""}
      className="main"
    />
  </FormGroup>
</Col>
```

```
        onChange={mainData}
    />
    <span className="text-danger">{iserror.date}</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="password" className="fw-600 fs-5">
        Password
    </Label>
    <Input
        id="password"
        name="password"
        placeholder=""
        type="password"
        value={obj.password || ""}
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">{iserror.password}</span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="confirmpassword" className="fw-600 fs-5">
        Confirm Password
    </Label>
    <Input
        id="confirmpassword"
        name="confirmpassword"
        placeholder=""
        type="password"
        value={obj.confirmpassword || ""}
        className="main"
        onChange={mainData}
    />
    <span className="text-danger">
        {iserror.confirmpassword}
    </span>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="phoneno" className="fw-600 fs-5">
        Mobile no
    </Label>
```

```
</Label>
<Input
    id="phoneno"
    name="phoneno"
    placeholder=""
    value={obj.phoneno || ""}
    type="number"
    className="main"
    onChange={mainData}
/>
{iserror.phoneno}
</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="exampleCheck4"
        name="language"
        type="checkbox"
        className="language me-2"
        value="react"
        onChange={mainData}
        checked={obj.language.includes("react")}

    >
    <Label check for="exampleCheck4" className="px-2">
        REACT
    </Label>
</div>
<span className="text-danger">{iserror.language}</span>
</Col>
<Col md={6}>
    <Label for="example" className="fw-600 fs-5">
        Gender
    </Label>
    <div className="d-flex">
        <div>
```

```
<Input
  id="exampleCheck5"
  name="gender"
  type="radio"
  className="gender me-2"
  value="male"
  onChange={mainData}
  checked={obj.gender === "male"}
/>
<Label check for="exampleCheck5" 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>
<span className="text-danger">{iserror.gender}</span>
</Col>
<Col md={12}>
  <FormGroup>
    <Label for="information" className="fw-600 fs-5">
      Information
    </Label>
    <Input
      id="information"
      name="information"
      placeholder=""
      value={obj.information || ""}
      type="textarea"
      className="main"
      onChange={mainData}
    />
    <span className="text-danger">{iserror.information}</span>
```

```

        </FormGroup>
    </Col>
</Row>
<div className="text-center">
    <button className="my-2 btn btn-secondary submit fs-4">
        Submit
    </button>
</div>
</Form>
</Container>
</Col>
</Row>
<Container>
    <MainTable
        mainArray={array}
        filterArray={filterArray}
        editfun={editfun}
        deletefun={deletefun}
    />
    </Container>
    </Fragment>
);
};

export default MainForm;

```

mantable.jsx

```

import React, { Fragment } from "react";
import { Table } from "reactstrap";

function MainTable(props) {
    return (
        <Fragment>
            <div className="my-5">
                <h1 className="text-center py-3">Employee Information</h1>
                <Table className="p-3">
                    <thead>
                        <tr>
                            <th>Sr No</th>
                            <th>Profile</th>
                            <th>Name</th>
                            <th>LastName</th>
                        </tr>
                    </thead>
                    <tbody>
                    </tbody>
                </Table>
            </div>
        </Fragment>
    );
}

export default MainTable;

```

```

        <th>Email</th>
        <th>Phone No</th>
        <th>Gender</th>
        <th>Language</th>
        <th>Action</th>
    </tr>
</thead>
<tbody>
    {/* main array ni jaygae direct filter array par map chalavyo che so
    jo koi filter check nai karyu hoy tyare badha undefined hase ane full array a
    filter array ma return aavse aane a table ma dekhase aane jyare filter karsu tyare
    to filter array j batavano che */}
    {/* {console.log(props.filterArray)} */}
    {props.filterArray.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.lastname}</td>
                <td>{x.email}</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;
```

Double Form

App.js

```
import 'bootstrap/dist/css/bootstrap.min.css';
import './App.css';
import FormComponent from './components/FormComponent';
import { createContext, useEffect, useState } from 'react';
import SecondFormComponent from './components/SecondFormComponent';
export let Cardcontext = createContext()
export let Cardcontextsecond = createContext()
function App() {
  let [array, setarray] = useState(JSON.parse(localStorage.getItem("array")) || [])
  let [count, setcount] = useState(JSON.parse(localStorage.getItem("count")) || 0)
  let [obj, setobj] = useState({ title: '', subtitle: '', image: '', information: '', button: '' })
  let [editobj, seteditobj] = useState({})
  let [editid, seteditid] = useState(0)

  let [arraysecond, setarraysecond] =
  useState(JSON.parse(localStorage.getItem("arraysecond")) || [])
  let [countsecond, setcountsecond] =
  useState(JSON.parse(localStorage.getItem("countsecond")) || 0)
  let [objsecond, setobjsecond] = useState({ name: '', surname: '', image: '' })
  let [editobjsecond, seteditobjsecond] = useState({})
  let [editidsecond, seteditidsecond] = useState(0)

  useEffect(() => {
    localStorage.setItem("array", JSON.stringify(array || []));
    localStorage.setItem("count", JSON.stringify(count || 0));
    localStorage.setItem("arraysecond", JSON.stringify(arraysecond || []));
    localStorage.setItem("countsecond", JSON.stringify(countsecond || 0));
  }, [array, count, arraysecond, countsecond])
```

```

useEffect(() => {
  setarray(JSON.parse(localStorage.getItem("array")))
  setcount(JSON.parse(localStorage.getItem("count")))
  setarraysecond(JSON.parse(localStorage.getItem("arraysecond")))
  setcountsecond(JSON.parse(localStorage.getItem("countsecond")))
}, [])
return (
  <div>
    <div className='container'>
      <div className='row'>
        <div className='col-6'>
          <Cardcontext.Provider value={{ array, count, obj, editobj, editid,
setarray, setcount, setobj, seteditobj, seteditid }} >
            <FormComponent />
          </Cardcontext.Provider>
        </div>
        <div className='col-6'>
          <Cardcontextsecond.Provider value={{ arraysecond, countsecond,
objsecond, editobjsecond, editidsecond, setarraysecond, setcountsecond,
setobjsecond, seteditobjsecond, seteditidsecond }} >
            <SecondFormComponent />
          </Cardcontextsecond.Provider>
        </div>
      </div>
    </div>
  );
}

export default App;

```

Form Component

```

import React, { useContext, useEffect, useRef, useState } from 'react'
import { Button, Col, Form, FormGroup, Input, Label, Row } from 'reactstrap'
import { Cardcontext } from '../App'
import CardComponent from './CardComponent'
const FormComponent = () => {
  let value = useContext(Cardcontext)
  const fileref = useRef()
  const getBase64 = (file) => new Promise(function (resolve, reject) {
    let reader = new FileReader();
    reader.readAsDataURL(file);
  })

```

```

        reader.onload = () => resolve(reader.result)
        reader.onerror = (error) => reject('Error: ', error);
    })
    const changedata = async (e) => {
        if (e.target.name == "image") {
            value.obj[e.target.name] = await getBase64(e.target.files[0])
        }
        else {
            value.obj[e.target.name] = e.target.value
        }
        value.setobj({ ...value.obj })
    }

    const save = (e) => {
        e.preventDefault()
        if (value.editid == 0) {
            value.count = value.count + 1
            value.obj.id = value.count
            console.log(value.obj)
            value.array.push(value.obj)
        }
        else {
            value.editid = 0;
            value.array.splice(value.array.findIndex(x => x.id == value.editid),
1, value.obj)
        }
        value.setobj({ ...value.obj })
        value.setarray([...value.array])
        value.setcount(value.count)
        value.setobj({})
        value.seteditid(value.editid)
        fileref.current.value = ""
    }

    return (
        <div>
            <div className="container text-bg-dark rounded-3 py-5 my-3">
                <h1 className='text-center py-3'>First Form</h1>
                <Form className=' '>
                    <Row>
                        <Col md={6}>
                            <FormGroup>
                                <Label for="title" className='fs-5'>
                                    Title
                                </Label>

```

```
<Input
      id="title"
      name="title"
      type="text"
      value={value.obj.title || ''}
      placeholder='Add Title'
      onChange={changedata}
    />
</FormGroup>
</Col>
<Col md={6}>
  <FormGroup>
    <Label for="subtitle" className='fs-5'>
      SubTitle
    </Label>
    <Input
      id="subtitle"
      name="subtitle"
      type="text"
      value={value.obj.subtitle || ''}
      placeholder="Sub Title"
      onChange={changedata}
    />
  </FormGroup>
</Col>
</Row>
<Row>
  <Col md={12}>
    <FormGroup>
      <Label for="image" className='fs-5'>
        Image
      </Label>
      <input
        id="image"
        name="image"
        type="file"
        placeholder='File'
        onChange={changedata}
        ref={fileref}
        className='form-control'
      />
    </FormGroup>
  </Col>
</Row>
```

```

<Row>
  <Col md={12}>
    <FormGroup>
      <Label for="information" className='fs-5'>
        Information
      </Label>
      <Input
        id="information"
        name="information"
        placeholder="Enter Description"
        type='textarea'
        value={value.obj.information || ""}
        onChange={changedata}
      />
    </FormGroup>
  </Col>
</Row>
<div className='text-center'>
  <Button className=' ' onClick={save}>
    Sign in
  </Button>
</div>
</Form>
</div>
<CardComponent value={value.array} >
  <thead>
    <tr>
      <th>
        Sr No.
      </th>
      <th>
        Title
      </th>
      <th>
        SubTitle
      </th>
      <th>
        Image
      </th>
      <th>
        Action
      </th>
    </tr>
  </thead>
</CardComponent >

```

```

        </div>
    )
}

export default FormComponent

```

Second Form Component

```

import React, { useContext, useEffect, useRef, useState } from 'react'
import { Button, Col, Form, FormGroup, Input, Label, Row } from 'reactstrap'
import validationdata from '../JSON/validation.json'
import { Cardcontextsecond } from '../App'
import CardComponent from './CardComponent'
const SecondFormComponent = () => {
    let value = useContext(Cardcontextsecond)
    console.log(value.countsecond)
    const fileref = useRef()
    let [errorobj, seterrorobj] = useState({})
    const getBase64 = (file) => new Promise(function (resolve, reject) {
        let reader = new FileReader();
        reader.readAsDataURL(file);
        reader.onload = () => resolve(reader.result)
        reader.onerror = (error) => reject('Error: ', error);
    })
    const changedata = async (e) => {
        if (e.target.name == "image") {
            value.objsecond[e.target.name] = await getBase64(e.target.files[0])
        }
        else {
            value.objsecond[e.target.name] = e.target.value
        }
        value.setobjsecond({ ...value.objsecond })
        // validate(e.target.name)
    }

    const save = (e) => {
        // for (let x of Object.keys(value.objsecond)) {
        //     validate(x)
        // }
        e.preventDefault()
        if (value.editidsecond == 0) {
            value.countsecond = value.countsecond + 1
            value.objsecond.id = value.countsecond
        }
    }
}

```

```

        console.log(value.objsecond)
        value.arraysecond.push(value.objsecond)
    }
    else {
        value.editidsecond = 0;
        value.arraysecond.splice(value.arraysecond.findIndex(x => x.id ==
value.editidsecond), 1, value.objsecond)
    }
    value.setobjsecond({ ...value.objsecond })
    value.setarraysecond([...value.arraysecond])
    value.setcountsecond(value.countsecond)
    value.setobjsecond({})
    value.seteditidsecond(value.editidsecond)
    console.log(value.arraysecond)
    fileref.current.value = ""
}

// const validate = (name) => {
//     let validationobj = validationdata.find((x) => x.name == name)
//     let validationerror = validationobj.conditions.find((x) =>
eval(x.condition))
//     if (validationobj) {
//         if (validationerror) {
//             errorobj[name] = validationerror.error
//         }
//         else {
//             delete errorobj[name]
//         }
//     }
// }
// }

return (
    <div>
        <div className="container text-bg-dark rounded-3 py-5 my-3">
            <h1 className='text-center py-3'>Second Form</h1>
            <Form className=' '>
                <Row>
                    <Col md={6}>
                        <FormGroup>
                            <Label for="name" className='fs-5'>
                                Name
                            </Label>
                            <Input
                                id="name"
                                name="name"
                                type="text"
                            </Input>
                        </FormGroup>
                    </Col>
                </Row>
            </Form>
        </div>
    </div>
)

```

```

        value={value.objsecond.name || ''}
        placeholder='Add Name'
        onChange={changedata}
      />
      <span>{errorobj.name}</span>
    </FormGroup>
  </Col>
  <Col md={6}>
    <FormGroup>
      <Label for="surname" className='fs-5'>
        Surname
      </Label>
      <Input
        id="surname"
        name="surname"
        type="text"
        value={value.objsecond.surname || ''}
        placeholder="Surname"
        onChange={changedata}
      />
      <span>{errorobj.surname}</span>
    </FormGroup>
  </Col>
</Row>
<Row>
  <Col md={12}>
    <FormGroup>
      <Label for="image" className='fs-5'>
        Image
      </Label>
      <input
        id="imagesecond"
        name="imagesecond"
        type="file"
        placeholder='File'
        onChange={changedata}
        ref={fileref}
        className='form-control'
      />
      <span>{errorobj.image}</span>
    </FormGroup>
  </Col>
</Row>
<div className='text-center'>
  <Button className=' ' onClick={save}>

```

```

        Sign in
      </Button>
    </div>
  </Form>
</div>
<CardComponent value={value.arraysecond}>
  <thead>
    <tr>
      <th>
        Sr no
      </th>
      <th>
        Name
      </th>
      <th>
        Surname
      </th>
      <th>
        Image
      </th>
      <th>
        Action
      </th>
    </tr>
  </thead>
</CardComponent>
</div>
)
}

export default SecondFormComponent

```

Card Component

```

import React, { useContext, useEffect, useState } from 'react'
import { Table } from 'reactstrap'

const CardComponent = (props) => {
  console.log(props)
  const deletefunction = (id) => {
    props.value.array.splice(props.value.array.findIndex(x => x.id == id), 1)
    props.value.setarray([...props.value.array])
  }
}

```

```
const editfunction = (id) => {
    props.value.seteditid(id)
    props.value.obj = props.value.array.find((x) => x.id == id)
    props.value.setobj({ ...props.value.obj })
}

return (
    <>
    <Table
        bordered
        dark
        responsive
        size=""
        striped
    >
        {props.children}
        <tbody>
            {props.value?.map((x, i) => {
                return (
                    <tr key={i}>
                        <th scope="row">
                            {i + 1}
                        </th>
                        <td>
                            {x.title || x.name}
                        </td>
                        <td>
                            {x.subtitle || x.surname}
                        </td>
                        <td>
                            <img src={x.image || x.imagesecond} style={{height: '50px', width: "50px", objectFit: 'cover'}} alt="" />
                        </td>
                        <td>
                            <button>Edit</button>
                            <button>Delete</button>
                        </td>
                    </tr>
                )
            ))}
        </tbody>
    </Table>
)
)
```

```
}
```

```
export default CardComponent
```

Function Api

```
import React, { Fragment, useEffect, useState } from 'react'
import axios from 'axios'
import { Col, Container, Form, FormGroup, Input, Label, Row, Table } from
'reactstrap'
const Api = () => {
  let [arr, setarr] = useState([])
  let [obj, setobj] = useState({ hobbies: '' })
  /* api methods are different type. Which is created by backend developer
  Mostly 4 to 5 api method available
  1=>getapi
  2=>getapi by id
  3=>post api
  4=>delete api
  5=>update api

  1=>getapi
  in this api we have link which we put in axios.get('url').then((res)=>
res) in response all data we get in res. if error occur then it is show in catch.
if api get run then it is show in inspect network in fetch/xhr show
  headers=>in which status code show. if 200 means run successfully if we
write wrong anything then 404 not found.
  that type of different status code available for network.
  response=>in response we get data from api
  payload=> in payload we see query which we pass for get data like id we
pass for data find
  preview=> in preview status preview

  2=>getapi by id in we have pass url with query means id or anything by which
api find data from database and get data
  url=>https://student-api.mycodelibraries.com/api/student/get-student-by-id?id=anyid like this

  3=>post api=>we have to pass obj which we want to set data in database
  4=>in delete api url is connect with query which we want to delete quer yis
any possible id or any key firstName lastName
  url=>https://student-api.mycodelibraries.com/api/student/delete?id=64ca1518378d0635bc1001fc
```

5=>update api we have to pass new obj which we want to replace with existing for that

that new obj pass with same id which is previous one
and for that formate given like in this api we have to pass obj id like this
obj.id not like this obj._id

```

sample object pass given in api
{
    "firstName": "Shaikhs",
    "lastName": "Irshad",
    "age": 1,
    "hobbies": "Reading,Exersice",
    "gender": "Male",
    "city": "Surat",
    "id": "612a05e155c2d838a4c62e5f"
}
*/

```

```

const setData = () => {
    axios.post('https://student-api.mycodelibraries.com/api/student/add',
obj)
    .then((res) => {
        console.log(res)
        getData()
    }
    )
    .catch((err) => console.log(err))
}
const getData = () => {
    axios.get('https://student-api.mycodelibraries.com/api/student/get')
    .then((res) => {
        arr = res.data.data
        setarr([...arr])
    }
    )
    .catch((err) => console.log(err))
}
const deleteapi = (a) => {
    a = `https://student-api.mycodelibraries.com/api/student/delete?id=${a}`
    axios.delete(a).then((res) => {
        getData()
    }).catch((err) => console.log(err))
}

const editFunction = (id) => {

```

```
        axios.get("https://student-api.mycodelibraries.com/api/student/get-student-by-id?id=" + id).then((res) => {
            obj = res.data.data
            obj.hobbies = obj.hobbies.split(",")
            setobj({ ...obj })
        }).catch((err) => { console.log(err) })
    }

    const updateapi = () => {
        obj.id = obj._id
        axios.post('https://student-api.mycodelibraries.com/api/student/update', obj).then((res) => getData()).catch((err) => console.log(err))
    }

    useEffect(() => {
        getData()
    }, [])

    const changeData = (e) => {
        if (e.target.name == "hobbies") {
            if (e.target.checked) {
                obj.hobbies = [...obj.hobbies, e.target.value]
            }
            else {
                obj.hobbies = obj.hobbies.filter((x) =>
!x.includes(e.target.value))
            }
        }
        else {
            obj[e.target.name] = e.target.value
        }
        setobj({ ...obj })
    }

    const submitFunction = (e) => {
        e.preventDefault();
        if (obj._id == undefined) {
            setData()
        }
        else {
            updateapi()
        }
        obj = { hobbies: '' }
        setobj({ ...obj })
    }
}
```

```

return (
  <div>
    <Row>
      <Col xs={6} className="offset-3">
        <Container className="mt-1 py-1 px-4 border border-1 border-
black rounded-2 shadow-lg">
          <h1 className="text-center py-3">Student Form</h1>
          <Form onSubmit={(e) => { submitFunction(e) }}>
            <Row>
              <Col md={6}>
                <FormGroup>
                  <Label for="firstName" className="fw-
600">
                    First Name
                  </Label>
                  <Input
                    id="firstName"
                    name="firstName"
                    placeholder=""
                    type="text"
                    className="main"
                    onChange={changeData}
                    value={obj.firstName || ''}
                  />
                </FormGroup>
              </Col>
              <Col md={6}>
                <FormGroup>
                  <Label for="lastName" className="fw-600
">
                    last Name
                  </Label>
                  <Input
                    id="lastName"
                    name="lastName"
                    placeholder=""
                    type="text"
                    className="main"
                    onChange={changeData}
                    value={obj.lastName || ''}
                  />
                </FormGroup>
              </Col>
              <Col md={6}>

```

```
<FormGroup>
  <Label for="age" className="fw-600 ">
    Age
  </Label>
  <Input
    id="age"
    name="age"
    placeholder=""
    type="number"
    className="main"
    onChange={changeData}
    value={obj.age || ''}

  />
</FormGroup>
</Col>
<Col md={6}>
  <FormGroup>
    <Label for="city" className="fw-600 ">
      City
    </Label>
    <select onChange={changeData}
value={obj.city || ''} name="city" className="form-select">
      <option value="surat">Surat</option>
      <option
value="bharuch">Bharuch</option>
      <option
value="vadodara">Vadodara</option>
    </select>
  </FormGroup>
</Col>
<Col md={6}>
  <Label for="example" className="fw-600 ">
    Gender
  </Label>
  <div className="d-flex">
    <div>
      <Input
        id="exampleCheck3"
        name="gender"
        type="radio"
        className="gender me-2"
        onChange={changeData}
        value="Male"


```

```
        checked={obj.gender == "Male" ||  
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"  
            onChange={changeData}  
            value="Female"  
            checked={obj.gender == "Female"  
|| obj.gender == "female"}  
        />  
  
        <Label  
            check  
            for="radio"  
            className="px-2"  
        >  
            Female  
        </Label>  
    </div>  
    </div>  
    </Col>  
    <Col md={12} className="">  
        <Label  
            check  
            for="example"  
            className="fw-600  
my-2"  
        >  
            hobbies  
        </Label>  
        <Row className="">  
            <Col xs={3}>  
                <Input
```

```
        id="Travelling"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Travelling"
        checked={obj.hobbies?.includes('Travelling')}

    />
    <Label
        check
        for="Travelling"
        className="px-2"
    >
        Travelling
    </Label>
</Col>
<Col xs={3}>
    <Input
        id="Reading"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Reading"
        checked={obj.hobbies?.includes('Reading')}

    />
    <Label
        check
        for="Reading"
        className="px-2"
    >
        Reading
    </Label>
</Col>
<Col xs={3}>
    <Input
        id="Exersice"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Exersice"
    >

```

```

        checked={obj.hobbies?.includes('E
xersice')}

        />
<Label
    check
    for="Exersice"
    className="px-2"
    >
    Exersice
</Label>
</Col>
</Row>
</Col>
</Row>
<div className="text-center">
    <button className="my-2 btn btn-secondary submit
fs-4">
        Submit
    </button>
</div>
</Form>
</Container>
</Col>
</Row>
<div className="container bg-body-secondary mt-3">
    <h2 className='text-center py-3'>Form</h2>
    <Table className="">
        <thead>
            <tr>
                <th>Sr No</th>
                <th>FirstName</th>
                <th>LastName</th>
                <th>Age</th>
                <th>City</th>
                <th>Gender</th>
                <th>Hobbies</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            {arr?.map((x, i) => {
                return <tr key={i + 1}>
                    <td>{i + 1}</td>
                    <td>{x.firstName}</td>
                    <td>{x.lastName}</td>
                    <td><button type="button" onClick={()=>
                        setArr(
                            arr?.filter((el) => el.id !== x.id)
                        )
                    }>Delete</button>
                    </td>
                </tr>
            })
        }
    </tbody>
</Table>
</div>

```

```

        <td>{x.age}</td>
        <td>{x.city}</td>
        <td>{x.gender}</td>
        <td>{x.hobbies}</td>
        <td>
            <button onClick={() => deleteapi(x._id)}>
                Delete
            </button>
            <button onClick={() => editFunction(x._id)}>
                Edit
            </button>
        </td>
    </tr>
)
)
}

export default Api

```

Class Api

```

import { Fragment, PureComponent } from "react";
import { Col, Container, Form, FormGroup, Input, Label, Row, Table } from
"reactstrap";
import axios from "axios";
class ClassApi extends PureComponent {
    constructor() {
        super()
        this.state = {
            obj: {},
            arr: []
        }
    }
    getapi = () => {
        axios.get('https://student-api.mycodelibraries.com/api/student/get')
            .then((res) =>
                this.setState({ arr: res.data.data })
            )
            .catch((error) => console.log(error))
    }
    postapi = () => {

```

```

        axios.post('https://student-api.mycodelibraries.com/api/student/add',
this.state.obj)
            .then((res) =>
                this.getapi()
            )
            .catch((error) => console.log(error))
        }
    componentDidMount() {
        this.getapi()
        // this.state.obj = { hobbies: [], firstName: '', lastName: '', gender: '',
        age: '', city: '' }
        // this.setState({ obj: this.state.obj })
    }
    save = (e) => {
        e.preventDefault()
        if (this.state.obj._id == undefined) {
            this.postapi()
        }
        else {
            console.log("updateapi")
            this.updateapi()
        }
        this.setState({ obj: {} })
    }
    changedata = (e) => {
        if (e.target.name == "hobbies") {
            // if (e.target.checked) {
            //     this.state.obj.hobbies.push(e.target.value)
            // }
            // else {
            //     this.state.obj.hobbies = this.state.obj.hobbies.filter((x) =>
!x.includes(e.target.value))
            // }
            if (e.target.checked === true) {
                this.state.obj.hobbies = [
                    ...this.state.obj.hobbies ? [...this.state.obj.hobbies] : [],
e.target.value
                ]
            }
            else {
                this.state.obj.hobbies = this.state.obj.hobbies.filter((x) => {
                    return x !== e.target.value
                })
            }
        }
    }
}

```



```
<Label for="firstName" className="fw-600">
    First Name
</Label>
<Input
    id="firstName"
    name="firstName"
    placeholder="First Name"
    type="text"
    className="main"
    onChange={this.changedata}
    value={this.state.obj.firstName ? this.state.obj.firstName : null}>
</>

</FormGroup>
</Col>
<Col md={6}>
    <FormGroup>
        <Label for="lastName" className="fw-600">
            last Name
        </Label>
        <Input
            id="lastName"
            name="lastName"
            placeholder="Last Name"
            type="text"
            className="main"
            onChange={this.changedata}
            value={this.state.obj.lastName ? this.state.obj.lastName : null}>
</>
    </FormGroup>
</Col>
<Col md={6}>
    <FormGroup>
        <Label for="age" className="fw-600">
            Age
        </Label>
        <Input
            id="age"
            name="age"
            placeholder="Age"
            type="number"
```

```
        className="main"
        value={this.state.obj.age ?
this.state.obj.age : null}
        onChange={this.changedata}
    />
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
<Label for="city" className="fw-600">
    City
</Label>
<select onChange={this.changedata}
        value={this.state.obj.city ?
this.state.obj.city : null}
        name="city" className="form-select">
<option
        value="surat">Surat</option>
<option
        value="bharuch">Bharuch</option>
<option
        value="vadodara">Vadodara</option>
    </select>
</FormGroup>
</Col>
<Col md={6}>
<Label for="example" className="fw-600 ">
    Gender
</Label>
<div className="d-flex">
    <div>
        <Input
            id="exampleCheck3"
            name="gender"
            type="radio"
            className="gender me-2"
            onChange={this.changedata}
            value="Male"
            checked={this.state.obj.gender?.includes("Male") || this.state.obj.gender?.includes("male")}
        />
        <Label
            check

```

```
        for="radio"
        className="px-2"
      >
        Male
      </Label>
    </div>
  <div>
    <Input
      id="exampleCheck3"
      name="gender"
      type="radio"
      className="gender me-2"
      onChange={this.changedata}
      value="Female"
      checked={this.state.obj.gender?
        .includes("Female") || this.state.obj.gender?.includes("female")}
    />
    <Label
      check
      for="radio"
      className="px-2"
    >
      Female
    </Label>
  </div>
</div>
</Col>
<Col md={12} className="">
  <Label
    check
    for="example"
    className="fw-600 my-2"
  >
    hobbies
  </Label>
<Row className="">
  <Col xs={3}>
    <Input
      id="Travelling"
      name="hobbies"
      type="checkbox"
      className="language me-2"
      onChange={this.changedata}
      value="Travelling"
    />
  </Col>
</Row>

```

```
        checked={this.state.obj.hobbies.includes('Travelling')}
```

```
    />
    <Label
      check
      for="Travelling"
      className="px-2"
    >
      Travelling
    </Label>
  </Col>
  <Col xs={3}>
    <Input
      id="Reading"
      name="hobbies"
      type="checkbox"
      className="language me-2"
      onChange={this.changedata}
      value="Reading"
      checked={this.state.obj.hobbies.includes('Reading')}
    />
    <Label
      check
      for="Reading"
      className="px-2"
    >
      Reading
    </Label>
  </Col>
  <Col xs={3}>
    <Input
      id="Exersice"
      name="hobbies"
      type="checkbox"
      className="language me-2"
      onChange={this.changedata}
      value="Exersice"
      checked={this.state.obj.hobbies.includes('Exersice')}
    />
    <Label
      check
      for="Exersice"
      className="px-2"
    >
```

```

        >
      Exersice
    </Label>
  </Col>
</Row>
</Col>
</Row>
<div className="text-center">
  <button className="my-2 btn btn-secondary
submit fs-4">
  Submit
</button>
</div>
</Form>
</Container>
</Col>
</Row>
<div className="container bg-body-secondary mt-3">
  <h2 className='text-center py-3'>Form</h2>
  <Table className="">
    <thead>
      <tr>
        <th>Sr No</th>
        <th>FirstName</th>
        <th>LastName</th>
        <th>Age</th>
        <th>City</th>
        <th>Gender</th>
        <th>Hobbies</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      {this.state.arr?.map((x, i) => {
        return <tr key={i + 1}>
          <td>{i + 1}</td>
          <td>{x.firstName}</td>
          <td>{x.lastName}</td>
          <td>{x.age}</td>
          <td>{x.city}</td>
          <td>{x.gender}</td>
          <td>{x.hobbies}</td>
          <td>
            <button onClick={() =>
              this.deleteapi(x._id)} className='me-2 btn text-bg-danger'>Delete</button>

```

```

                <button onClick={() =>
this.editFunction(x._id)} className='btn text-bg-warning'>Edit</button>
            </td>
        </tr>
    )}
</tbody>
</Table>
</div>
</Fragment>
)
}
}
export default ClassApi

```

Fetch Api

```

import React, { useEffect, useRef, useState } from "react";
import {
  Col,
  Container,
  Form,
  FormGroup,
  Input,
  Label,
  Row,
  Table,
} from "reactstrap";
const FetchApi = () => {
  let [arr, setarr] = useState([]);
  let [obj, setobj] = useState({ hobbies: "" });
  let reference = useRef();

  const setData = () => {
    let formdata = new FormData();
    formdata.append("userImage", obj.userImage);
    formdata.append("firstName", obj.firstName);
    formdata.append("lastName", obj.lastName);
    formdata.append("age", obj.age);
    formdata.append("city", obj.city);
    formdata.append("gender", obj.gender);
    formdata.append("hobbies", obj.hobbies);
    /* if we want to send data in api with image also then we have to use
    formdata which is method of javascript */
  }
}

```

```

/* in fetch api we have to send data by body key and we have to write
method..
    if we have to use protected api then we have to send headers with token

    Important=> in fetch api response . if we want data then we have to use
res.json() in which we get data and for that we have to use async await and then
we have to store any state like
    a=await res.json()
    then arr=a.data
    if we send object during post method we have to data in stringify format and
if we send data in formdata then we dont have to stringify it

    body:JSON.stringify(obj)*/

fetch(
  "https://student-api.mycodelibraries.com/api/user/add", {
    method: 'POST',
    body: formdata
})
  .then((res) => {
    getData();
  })
  .catch((err) => console.log(err));
};

const getData = () => {
  /* fetch is method of javascript */
  fetch("https://student-api.mycodelibraries.com/api/user/get")
    .then(async (res) => {
      arr = await res.json();
      setarr([...arr.data]);
    })
    .catch((err) => console.log(err));
};

const deleteapi = (id) => {
  fetch('https://student-api.mycodelibraries.com/api/user/delete?id=' + id, {
    method: 'DELETE',
  })
    .then((res) => {
      getData();
    })
    .catch((err) => console.log(err));
};

const editFunction = (id) => {
  fetch(
    "https://student-api.mycodelibraries.com/api/user/get-user-by-id?id=" + id
  )
}

```

```

)
.then(async (res) => {
  let a = await res.json()
  obj = a.data
  obj.hobbies = obj.hobbies.split(",")
  setobj({ ...obj });
})
.catch((err) => {
  console.log(err);
});
};

/* in this api when we update we have to pass id directly not like this _id
therefore we have to change key name from _id to id */
const updateapi = () => {
  let formdata1 = new FormData();
  formdata1.append("id", obj._id);
  formdata1.append("userImage", obj.userImage);
  formdata1.append("firstName", obj.firstName);
  formdata1.append("lastName", obj.lastName);
  formdata1.append("age", obj.age);
  formdata1.append("city", obj.city);
  formdata1.append("gender", obj.gender);
  formdata1.append("hobbies", obj.hobbies);
  fetch(`https://student-api.mycodelibraries.com/api/user/update?id=${obj._id}`,
}
  method: 'POST',
  body: formdata1
})
.then((res) => {
  console.log(res)
  getData()
})
.catch((err) => console.log(err));
};

useEffect(() => {
  getData();
}, []);

const changeData = (e) => {
  if (e.target.name === "hobbies") {
    if (e.target.checked) {
      obj.hobbies = [...obj.hobbies, e.target.value];
    } else {
      obj.hobbies = obj.hobbies.filter((x) => !x.includes(e.target.value));
    }
  }
};

```

```

        }
    } else if (e.target.name === "userImage") {
        obj[e.target.name] = e.target.files[0];
    } else {
        obj[e.target.name] = e.target.value;
    }
    setobj({ ...obj });
};

const submitFunction = (e) => {
    e.preventDefault();
    if (obj._id === undefined) {
        setData();
    } else {
        updateapi();
    }
    obj = { hobbies: "" };
    setobj({ ...obj });
    reference.current.value = "";
};
return (
    <div>
        <Row>
            <Col xs={6} className="offset-3">
                <Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
                    <h1 className="text-center py-3">Student Form</h1>
                    <Form
                        onSubmit={(e) => {
                            submitFunction(e);
                        }}
                    >
                    <Row>
                        <Col md={6}>
                            <FormGroup>
                                <Label for="firstName" className="fw-600">
                                    First Name
                                </Label>
                                <Input
                                    id="firstName"
                                    name="firstName"
                                    placeholder=""
                                    type="text"
                                    className="main"
                                    onChange={changeData}
                                >
                            </FormGroup>
                        </Col>
                    </Row>
                </Container>
            </Col>
        </Row>
    </div>
);

```

```
        value={obj.firstName || ""}
    />
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="lastName" className="fw-600 ">
        last Name
    </Label>
    <Input
        id="lastName"
        name="lastName"
        placeholder=""
        type="text"
        className="main"
        onChange={changeData}
        value={obj.lastName || ""}
    />
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="age" className="fw-600 ">
        Age
    </Label>
    <Input
        id="age"
        name="age"
        placeholder=""
        type="number"
        className="main"
        onChange={changeData}
        value={obj.age || ""}
    />
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
    <Label for="city" className="fw-600 ">
        City
    </Label>
    <select
        onChange={changeData}
        value={obj.city || ""}
        name="city"
    />
```

```
        className="form-select"
      >
        <option value="surat">Surat</option>
        <option value="bharuch">Bharuch</option>
        <option value="vadodara">Vadodara</option>
      </select>
    </FormGroup>
  </Col>
  <Col md={6}>
    <Label for="example" className="fw-600 ">
      Gender
    </Label>
    <div className="d-flex">
      <div>
        <Input
          id="exampleCheck3"
          name="gender"
          type="radio"
          className="gender me-2"
          onChange={changeData}
          value="Male"
          checked={obj.gender === "Male" || 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"
          onChange={changeData}
          value="Female"
          checked={
            obj.gender === "Female" || obj.gender === "female"
          }
        />
        <Label check for="radio" className="px-2">
          Female
        </Label>
      </div>
    </div>
  </Col>

```

```
</Col>
<Col md={12} className="">
  <Label
    check
    for="example"
    className="fw-600
      my-2"
  >
    hobbies
  </Label>
  <Row className="">
    <Col xs={3}>
      <Input
        id="Travelling"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Travelling"
        checked={obj.hobbies?.includes("Travelling")}
      />
      <Label check for="Travelling" className="px-2">
        Travelling
      </Label>
    </Col>
    <Col xs={3}>
      <Input
        id="Reading"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Reading"
        checked={obj.hobbies?.includes("Reading")}
      />
      <Label check for="Reading" className="px-2">
        Reading
      </Label>
    </Col>
    <Col xs={3}>
      <Input
        id="Exersice"
        name="hobbies"
        type="checkbox"
        className="language me-2"
      />
```

```
        onChange={changeData}
        value="Exersice"
        checked={obj.hobbies?.includes("Exersice")}
    />
    <Label check for="Exersice" className="px-2">
        Exersice
    </Label>
</Col>
</Row>
<Row className="py-2">
    <Col xs={12}>
        <Label check for="Exersice" className="py-2">
            Profile
        </Label>
        <input
            id="userImage"
            name="userImage"
            type="file"
            className="language me-2 form-control"
            onChange={changeData}
            ref={reference}
        />
    </Col>
</Row>
</Col>
</Row>
<div className="text-center">
    <button className="my-2 btn btn-secondary submit fs-4">
        Submit
    </button>
</div>
</Form>
</Container>
</Col>
</Row>
<div className="container bg-body-secondary mt-3">
    <h2 className="text-center py-3">Form</h2>
    <Table className="">
        <thead>
            <tr>
                <th>Sr No</th>
                <th>Profile</th>
                <th>FirstName</th>
                <th>LastName</th>
                <th>Age</th>
```

```

        <th>City</th>
        <th>Gender</th>
        <th>Hobbies</th>
        <th>Action</th>
    </tr>
</thead>
<tbody>
{arr?.map((x, i) => {
    return (
        <tr key={i + 1}>
            <td>{i + 1}</td>
            <td>
                <img src={x.image} alt="" width={40} height={40} />
            </td>
            <td>{x.firstName}</td>
            <td>{x.lastName}</td>
            <td>{x.age}</td>
            <td>{x.city}</td>
            <td>{x.gender}</td>
            <td>{x.hobbies}</td>
            <td>
                <button
                    onClick={() => deleteapi(x._id)}
                    className="me-2 btn text-bg-danger"
                >
                    Delete
                </button>
                <button
                    onClick={() => editFunction(x._id)}
                    className="btn text-bg-warning"
                >
                    Edit
                </button>
            </td>
        </tr>
    );
})}
</tbody>
</Table>
</div>
</div>
);
};

export default FetchApi;

```

Token Based api

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Access Tokens | GO REST". The page content is titled "Access Tokens" and includes a note: "These are your personal access tokens, do not share them on public websites." A green button labeled "New Access Token" is visible. Below is a table:

| Label | Token | Expires | Limit | Remaining | Action |
|---------------|--|---------|-------|-----------|--------|
| Primary Token | 8c4b9e86d5eb981e06cbcd32fe3e31a40a5337d7e3c4af1c018b11e32db4862a | Never | 90 | 90 | |

At the bottom, there are links for "Contact Us" and "Privacy Policy". The browser's status bar shows the date and time: "6:46 PM 8/8/2023".

```
import React, { useEffect, useRef, useState } from 'react'
import axios from 'axios'
import { Col, Container, Form, FormGroup, Input, Label, Row, Table } from
'reactstrap'
const TokenApi = () => {
  let [arr, setarr] = useState([])
  let [obj, setobj] = useState({})
  let token = {
    headers: {
      "Authorization": 'Bearer
8c4b9e86d5eb981e06cbcd32fe3e31a40a5337d7e3c4af1c018b11e32db4862a'
    }
  }
  /* in token based api we have to make object of token . it is based on api.
We can also make token of id password
  if we want data then we have to send token as a parameter
```

```
for this api we have make token from website gorest and login with gmail or
any email and take token
    token send in 3rd parameter in post method and as a 2nd parameter in delete
and get api
    if we not send token during get api then only that data access which is
access by any one
    if token not true or invalid then error 401 occur */
const setData = () => {
    axios.post('https://gorest.co.in/public/v2/users', obj, token)
        .then((res) => {
            console.log(res)
            getData()
        })
        .catch((err) => console.log(err))
}
const getData = () => {
    axios.get('https://gorest.co.in/public/v2/users', token)
        .then((res) => {
            arr = res.data
            setarr([...arr])
        })
        .catch((err) => console.log(err))
}
const deleteapi = (a) => {
    axios.delete('https://gorest.co.in/public/v2/users/' + a,
token).then((res) => {
        getData()
    }).catch((err) => console.log(err))
}

const editFunction = (id) => {
    obj = arr.find((x) => x.id == id)
    setobj({ ...obj })
}

const updateapi = () => {
    axios.patch('https://student-api.mycodelibraries.com/api/user/update',
obj).then((res) => getData()).catch((err) => console.log(err))
}

useEffect(() => {
    getData()
}, [])
```

```
const changeData = (e) => {
  obj[e.target.name] = e.target.value
  setobj({ ...obj })
}

const submitFunction = (e) => {
  e.preventDefault();
  if (obj.id === undefined) {
    setData()
  }
  else {
    updateapi()
  }
  obj = {}
  setobj({ ...obj })
}
return (
  <div>
    <Row>
      <Col xs={6} className="offset-3">
        <Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
          <h1 className="text-center py-3">Student Form</h1>
          <Form onSubmit={(e) => { submitFunction(e) }}>
            <Row>
              <Col md={6}>
                <FormGroup>
                  <Label for="name" className="fw-600">
                    Name
                  </Label>
                  <Input
                    id="name"
                    name="name"
                    placeholder=""
                    type="text"
                    className="main"
                    onChange={changeData}
                    value={obj.name || ''}
                  />
                </FormGroup>
              </Col>
              <Col md={6}>
                <FormGroup>
                  <Label for="email" className="fw-600 ">
```

```
        Email
    </Label>
    <Input
        id="email"
        name="email"
        placeholder=""
        type="email"
        className="main"
        onChange={changeData}
        value={obj.email || ''}
    />
</FormGroup>
</Col>

<Col md={6}>
    <Label for="example" className="fw-600 ">
        Gender
    </Label>
    <div className="d-flex">
        <div>
            <Input
                id="exampleCheck3"
                name="gender"
                type="radio"
                className="gender me-2"
                onChange={changeData}
                value="Male"
                checked={obj.gender === "Male" ||
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"
            />
            <Label
                check
                for="radio"
                className="px-2"
            >
                Female
            </Label>
        </div>
    </div>
</Col>
```

```
        onChange={changeData}
        value="Female"
        checked={obj.gender === "Female"
|| obj.gender === "female"}
      />

      <Label
        check
        for="radio"
        className="px-2"
      >
        Female
      </Label>
    </div>
  </div>
</Col>
<Col md={6}>
  <Label for="example" className="fw-600">
    Status
  </Label>
  <div className="d-flex">
    <div>
      <Input
        id="active"
        name="status"
        type="radio"
        className="status me-2"
        onChange={changeData}
        value="active"
        checked={obj.status === "active"
|| obj.status === "male"}
      />
      <Label
        check
        for="active"
        className="px-2"
      >
        Active
      </Label>
    </div>
    <div>
      <Input
        id="exampleCheck3"
        name="status"
        type="radio"
      />
    </div>
  </div>
</Col>
```

```

        className="status me-2"
        onChange={changeData}
        value="inactive"
        checked={obj.gender === "Inactive"
|| obj.gender === "inactive"}
      />
      <Label
        check
        for="radio"
        className="px-2"
      >
        Inactive
      </Label>
    </div>
  </div>
</Col>
</Row>
<div className="text-center">
  <button className="my-2 btn btn-secondary submit
fs-4">
    Submit
  </button>
</div>
</Form>
</Container>
</Col>
</Row>
<div className="container bg-body-secondary mt-3">
  <h2 className='text-center py-3'>Form</h2>
  <Table className="">
    <thead>
      <tr>
        <th>Sr No</th>
        <th>Name</th>
        <th>Email</th>
        <th>Gender</th>
        <th>Status</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      {arr?.map((x, i) => {
        return <tr key={i + 1}>
          <td>{x.id}</td>
          <td>{x.name}</td>
          <td>{x.email}</td>
          <td>{x.gender}</td>
          <td>{x.status}</td>
          <td><button
            className="my-2 btn btn-primary
            edit
            fs-4">
            Edit
          </button>
          <button
            className="my-2 btn btn-danger
            delete
            fs-4">
            Delete
          </button>
        </td>
      </tr>
    })
  </tbody>
</Table>
</div>

```

```

                <td>{x.email}</td>
                <td>{x.gender}</td>
                <td>{x.status}</td>
                <td>
                    <button onClick={() => deleteapi(x.id)}>
className='me-2 btn text-bg-danger'>Delete</button>
                    <button onClick={() => editFunction(x.id)}>
className='btn text-bg-warning'>Edit</button>
                </td>
            </tr>
        )}
    </tbody>
</Table>
</div>
</div>
)
}

export default TokenApi

```

User api

```

import React, { useEffect, useRef, useState } from 'react'
import axios from 'axios'
import { Col, Container, Form, FormGroup, Input, Label, Row, Spinner, Table } from 'reactstrap'
const UserApi = () => {
    let [arr, setarr] = useState([])
    let [obj, setobj] = useState({ hobbies: '' })
    let [rowdelete, setrowdelete] = useState(false)
    let reference = useRef()
    let [deleteid, setdeleteid] = useState(null)
    let [loading, setLoading] = useState(false)
    const [apicall, setapicall] = useState(false)

    const setData = () => {
        let formdata = new FormData()
        formdata.append('userImage', obj.userImage)
        formdata.append('firstName', obj.firstName)
        formdata.append('lastName', obj.lastName)
        formdata.append('age', obj.age)
        formdata.append('city', obj.city)
        formdata.append('gender', obj.gender)
    }
}

```

```
formdata.append('hobbies', obj.hobbies)
for (let x of formdata.entries()) {
    console.log(x)
}
setapicall(true)
setTimeout(() => {
    axios.post('https://student-api.mycodelibraries.com/api/user/add',
formdata)
    .then((res) => {
        console.log(res)
        getData()
        setapicall(false)
    })
    .catch((err) => console.log(err))
}, 5000);
}

const getData = () => {
    setapicall(true)
    axios.get('https://student-api.mycodelibraries.com/api/user/get')
        .then((res) => {
            arr = res.data.data
            setarr([...arr])
            setapicall(false)
        })
        .catch((err) => console.log(err))
}
const deleteapi = (id) => {
    setdeleteid(id)
    setrowdelet(true)
}
useEffect(() => {
    if (loading) {
        setapicall(true)
        axios.delete(`https://student-
api.mycodelibraries.com/api/user/delete?id=${deleteid}`).then((res) => {
            setrowdelet(false)
            getData()
            setapicall(false)
        }).catch((err) => {
            console.log(err)
            setrowdelet(false)
        })
    }
}
```

```
        else {
            setrowdelete(false)
        }
    }, [loading])

const Delete = () => {
    setLoading(true)
}
const NoDelete = () => {
    setLoading(false)
}
const editFunction = (id) => {
    setapicall(true)
    axios.get("https://student-api.mycodelibraries.com/api/user/get-user-by-
id?id=" + id).then((res) => {
        obj = res.data.data
        obj.hobbies = obj.hobbies.split(",")
        setobj({ ...obj })
        setapicall(false)
    }).catch((err) => { console.log(err) })
}

const updateapi = () => {
    obj.id = obj._id
    setapicall(true)
    axios.post('https://student-api.mycodelibraries.com/api/user/update' ,
obj).then((res) => {
        getData()
        setapicall(false)
    }).catch((err) => console.log(err))
}

useEffect(() => {
    getData()
}, [])

const changeData = (e) => {
    if (e.target.name === "hobbies") {
        if (e.target.checked) {
            obj.hobbies = [...obj.hobbies, e.target.value]
        }
        else {
            obj.hobbies = obj.hobbies.filter((x) =>
!x.includes(e.target.value))
        }
    }
}
```

```

        }
        else if (e.target.name === 'userImage') {
            obj[e.target.name] = e.target.files[0]
        }
        else {
            obj[e.target.name] = e.target.value
        }
        setobj({ ...obj })
    }

    const submitFunction = (e) => {
        e.preventDefault();
        if (obj._id === undefined) {
            setData()
        }
        else {
            updateapi()
        }
        obj = { hobbies: '' }
        setobj({ ...obj })
        reference.current.value = ''
    }
    return (
        <div>
            {rowdelet && <div className='parent'>
                <div>
                    <div className='pop-up'>
                        <h5><b>Are You Sure You Want to Delete?</b></h5>
                        <div className='d-flex justify-content-end'>
                            <button className='btn btn-danger me-2' onClick={Delete}>OK</button>
                            <button onClick={NoDelete} className='btn btn-warning'>Cancel</button>
                        </div>
                    </div>
                </div>
            }
            {apicall && <div className='loader position-fixed vh-100 vw-100 d-flex justify-content-center align-items-center'><img src={require('../assets/images/1200x1200.gif')} alt="" style={{ width: '200px', height: '200px' }} /></div>}

            <Row>
                <Col xs={6} className="offset-3">

```

```
<Container className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">

    <h1 className="text-center py-3">Student Form</h1>
    <Form onSubmit={(e) => { submitFunction(e) }}>
        <Row>
            <Col md={6}>
                <FormGroup>
                    <Label for="firstName" className="fw-600">
                        First Name
                    </Label>
                    <Input
                        id="firstName"
                        name="firstName"
                        placeholder=""
                        type="text"
                        className="main"
                        onChange={changeData}
                        value={obj.firstName || ''}
                    />
                </FormGroup>
            </Col>
            <Col md={6}>
                <FormGroup>
                    <Label for="lastName" className="fw-600">
                        last Name
                    </Label>
                    <Input
                        id="lastName"
                        name="lastName"
                        placeholder=""
                        type="text"
                        className="main"
                        onChange={changeData}
                        value={obj.lastName || ''}
                    />
                </FormGroup>
            </Col>
            <Col md={6}>
                <FormGroup>
                    <Label for="age" className="fw-600">
                        Age
                    </Label>
                </FormGroup>
            </Col>
        </Row>
    </Form>
</Container>
```

```
</Label>
<Input
    id="age"
    name="age"
    placeholder=""
    type="number"
    className="main"
    onChange={changeData}
    value={obj.age || ''}
  />
</FormGroup>
</Col>
<Col md={6}>
  <FormGroup>
    <Label for="city" className="fw-600 ">
      City
    </Label>
    <select onChange={changeData}
value={obj.city || ''} name="city" className="form-select">
      <option value="surat">Surat</option>
      <option
value="bharuch">Bharuch</option>
      <option
value="vadodara">Vadodara</option>
    </select>
  </FormGroup>
</Col>
<Col md={6}>
  <Label for="example" className="fw-600 ">
    Gender
  </Label>
  <div className="d-flex">
    <div>
      <Input
        id="exampleCheck3"
        name="gender"
        type="radio"
        className="gender me-2"
        onChange={changeData}
        value="Male"
        checked={obj.gender === "Male" ||
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"
        onChange={changeData}
        value="Female"
        checked={obj.gender === "Female"
|| obj.gender === "female"}
      />

      <Label
        check
        for="radio"
        className="px-2"
      >
        Female
      </Label>
    </div>
  </Col>
<Col md={12} className="">
  <Label
    check
    for="example"
    className="fw-600
my-2"
  >
    hobbies
  </Label>
<Row className="">
  <Col xs={3}>
    <Input
      id="Travelling"
      name="hobbies"
      type="checkbox"
      className="language me-2"
      onChange={changeData}
    >
  </Col>
  <Col xs={3}>
    <Input
      id="Sports"
      name="hobbies"
      type="checkbox"
      className="language me-2"
      onChange={changeData}
    >
  </Col>
  <Col xs={3}>
    <Input
      id="Reading"
      name="hobbies"
      type="checkbox"
      className="language me-2"
      onChange={changeData}
    >
  </Col>
</Row>

```

```
        value="Travelling"
        checked={obj.hobbies?.includes('T
ravelling')}

    />
    <Label
        check
        for="Travelling"
        className="px-2"
    >
        Travelling
    </Label>
</Col>
<Col xs={3}>
    <Input
        id="Reading"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Reading"
        checked={obj.hobbies?.includes('R
eading')}
    />
    <Label
        check
        for="Reading"
        className="px-2"
    >
        Reading
    </Label>
</Col>
<Col xs={3}>
    <Input
        id="Exersice"
        name="hobbies"
        type="checkbox"
        className="language me-2"
        onChange={changeData}
        value="Exersice"
        checked={obj.hobbies?.includes('E
xersice')}
    />
    <Label
        check
        for="Exersice"
    >
```

```

                className="px-2"
            >
            Exersice
        </Label>
    </Col>
</Row>
<Row className='py-2'>
    <Col xs={12}>
        <Label
            check
            for="Exersice"
            className="py-2"
        >
            Profile
        </Label>
        <input
            id="userImage"
            name="userImage"
            type="file"
            className="language me-2 form-
control"
            onChange={changeData}
            ref={reference}
        />
    </Col>
</Row>
</Col>
</Row>
<div className="text-center">
    <button className="my-2 btn btn-secondary submit
fs-4">
        Submit
    </button>
</div>
</Form>
</Container>
</Col>
</Row>

<div className="container bg-body-secondary mt-3">
    <h2 className='text-center py-3'>Form</h2>
    <Table className="">
        <thead>
            <tr>
                <th>Sr No</th>

```

```

        <th>Profile</th>
        <th>FirstName</th>
        <th>LastName</th>
        <th>Age</th>
        <th>City</th>
        <th>Gender</th>
        <th>Hobbies</th>
        <th>Action</th>
    </tr>
</thead>
<tbody>
    {arr?.map((x, i) => {
        return <tr key={i + 1}>
            <td>{i + 1}</td>
            <td>
                <img src={x.image} alt="" width={40}
height={40} />
            </td>
            <td>{x.firstName}</td>
            <td>{x.lastName}</td>
            <td>{x.age}</td>
            <td>{x.city}</td>
            <td>{x.gender}</td>
            <td>{x.hobbies}</td>
            <td>
                <button onClick={() => deleteapi(x._id)}>Delete</button>
                <button onClick={() => editFunction(x._id)}>Edit</button>
            </td>
        </tr>
    ))}
</tbody>
</Table>
</div>
</div>
)
}

export default UserApi

```

Product api

```
import React, { Component, Fragment } from 'react'
import { Button, Card, CardBody, CardSubtitle, CardText, CardTitle, Col, Container, Form, FormGroup, Input, Label, Row, Table } from 'reactstrap'
import axios from 'axios'
import { FiEdit } from 'react-icons/fi'
import { MdDeleteSweep } from 'react-icons/md'
class ProductApi extends Component {
  constructor() {
    super()
    this.state = {
      obj: {},
      arr: []
    }
  }
  save = (e) => {
    e.preventDefault()
    if (this.state.obj._id === undefined) {
      this.postapi()
    }
    else {
      this.updateapi()
    }
  }
  componentDidMount() {
    this.getapi()
  }
  getapi = () => {
    axios.get('https://student-api.mycodelibraries.com/api/product/get')
      .then((res) => {
        this.state.arr = res.data.data
        this.setState({ arr: this.state.arr })
      })
      .catch((err) => { })
  }
  postapi = () => {
    axios.post('https://student-api.mycodelibraries.com/api/product/add',
    this.state.obj)
      .then((res) => {
        this.getapi()
        this.setState({ obj: {} })
      })
      .catch((err) => { })
  }
}
```

```

        }
        updateapi = () => {
            this.state.obj.id = this.state.obj._id;
            axios.post('https://student-api.mycodelibraries.com/api/product/update',
this.state.obj).then((res) => {
                this.getapi()
                this.setState({ obj: {} })
            }).catch((err) => { })
        }
        editFunction = (id) => {
            axios.get('https://student-api.mycodelibraries.com/api/product/get-
product-by-id?id=' + id).then((res) => {
                res.data.data.clothSize = res.data.data.clothSize.split(",")
                this.setState({ obj: { ...res.data.data } })
            })
            .catch((err) => { })
        }
        deleteapi = (id) => {
            axios.delete('https://student-
api.mycodelibraries.com/api/product/delete?id=' + id).then((res) => {
this.getapi() }).catch((err) => { })
        }
        changeData = (e) => {
            if (e.target.name === 'clothSize') {
                if (e.target.checked) {
                    this.state.obj.clothSize =
                        this.state.obj.clothSize ? [...this.state.obj.clothSize,
e.target.value] : [e.target.value]
                }
                else {
                    this.state.obj.clothSize = this.state.obj.clothSize.filter((x) =>
x !== e.target.value)
                }
            }
            else {
                this.state.obj[e.target.name] = e.target.value
            }
            this.setState({ obj: { ...this.state.obj } })
        }
        render() {
            return (
                <div>
                    <Row>
                        <Col xs={12} sm={8} md={6} className="offset-sm-3">

```

```

<Container fluid='sm' className="mt-1 py-1 px-4 border border-1 border-black rounded-2 shadow-lg">
    <h1 className="text-center py-3">Student Form</h1>
    <Form onSubmit={this.save}>
        <Row>
            <Col md={6}>
                <FormGroup>
                    <Label for="Category" className="fw-600">
                        Category
                    </Label>
                    <select id='Category' className='form-select' name='category' onChange={this.changeData}>
                        <option value='boys' selected={this.state.obj.category?.includes('boys')}>Boys</option>
                        <option value='girls' selected={this.state.obj.category?.includes('girls')}>Girls</option>
                        <option value='trans' selected={this.state.obj.category?.includes('trans')}>Trans</option>
                    </select>
                </FormGroup>
            </Col>
            <Col md={6}>
                <FormGroup>
                    <Label for="productName" className="fw-600">
                        Product Name
                    </Label>
                    <Input id="productName" name="productName" placeholder="Product Name" type="text" className="main" onChange={this.changeData} value={this.state.obj.productName || ''}>
                    />
                </FormGroup>
            </Col>
            <Col md={6}>
                <FormGroup>

```

```
        <Label for="description"
      className="fw-600 ">
          Description
        </Label>
        <Input
          id="description"
          name="description"
          placeholder="description"
          type="text"
          className="main"
          onChange={this.changeData}
          value={this.state.obj.description
      || ''}
        />
      </FormGroup>
    </Col>
    <Col md={6}>
      <FormGroup>
        <Label for="price" className="fw-600">
          Price
        </Label>
        <Input
          id="price"
          name="price"
          placeholder="Price"
          type="number"
          className="main"
          onChange={this.changeData}
          value={this.state.obj.price ||
      ''}
        />
      </FormGroup>
    </Col>

    <Col md={6}>
      <Label for="example" className="fw-600 ">
          Stock Status
        </Label>
      <div className="d-flex">
        <div>
          <Input
            id="radioone"
            name="inStock"
            type="radio"

```

```
        className="gender me-2"
        value='inStock'
        onChange={this.changeData}
        checked={this.state.obj.inStock?.includes('inStock') || false}
      />
      <Label
        check
        for="radioone"
        className="px-2"
      >
        In Stock
      </Label>
    </div>
    <div>
      <Input
        id="radiotwo"
        name="inStock"
        type="radio"
        className="gender me-2"
        onChange={this.changeData}
        value='notInStock'
        checked={this.state.obj.inStock?.includes('notInStock') || false}
      />
      <Label
        check
        for="radiotwo"
        className="px-2"
      >
        Not in Stock
      </Label>
    </div>
  </Col>
<Col md={12} className="">
  <Label
    check
    for="example"
    className="fw-600 my-2"
  >
    Cloth Size
  </Label>
  <Row className="">
    <Col xs={6} md={2}>
```

```
<Input
      id="XXL"
      name="clothSize"
      type="checkbox"
      className="me-2"
      value='XXL'
      onChange={this.changeData}
      checked={this.state.obj.cloth
Size?.includes('XXL')}
    />
<Label
  check
  for="XXL"
  className="px-2"
>
  XXL
</Label>
</Col>

<Col xs={6} md={2}>
  <Input
    id="XL"
    name="clothSize"
    type="checkbox"
    className="me-2"
    value='XL'
    onChange={this.changeData}
    checked={this.state.obj.cloth
Size?.includes('XL')}
  />
  <Label
    check
    for="XL"
    className="px-2"
  >
    XL
  </Label>
</Col>

<Col xs={6} md={2}>
  <Input
    id="L"
    name="clothSize"
    type="checkbox"
    className="me-2"
  />
```

```
        value='L'
        onChange={this.changeData}
        checked={this.state.obj.cloth
Size?.includes('L')}

      />
      <Label
        check
        for="L"
        className="px-2"
      >
        L
      </Label>
    </Col>
    <Col xs={6} md={2}>
      <Input
        id="M"
        name="clothSize"
        type="checkbox"
        className="me-2"
        value='M'
        onChange={this.changeData}
        checked={this.state.obj.cloth
Size?.includes('M')}

      />
      <Label
        check
        for="M"
        className="px-2"
      >
        M
      </Label>
    </Col>

    <Col xs={6} md={2}>
      <Input
        id="S"
        name="clothSize"
        type="checkbox"
        className="me-2"
        value='S'
        onChange={this.changeData}
        checked={this.state.obj.cloth
Size?.includes('S')}>
```

```

        />
      <Label
        check
        for="S"
        className="px-2"
      >
        S
      </Label>
    </Col>
  </Row>
</Col>
</Row>
<div className="text-center">
  <button className="my-2 btn btn-secondary
submit fs-4">
    Submit
  </button>
</div>
</Form>
</Container>
</Col>
</Row>
<div className="container bg-body-secondary mt-3">
  <h2 className='text-center py-3'>Products</h2>
  <Row className='g-2'>
    {this.state.arr.map((x, i) => {
      return (
        <Col xs={4} key={i}>
          <Card>
            <div className='d-flex justify-content-
between my-2 card-header'>
              <MdDeleteSweep className='me-2 fs-1
text-bg-danger rounded' onClick={() => this.deleteapi(x._id)} />
              <FiEdit className='fs-2 text-bg-
warning rounded' onClick={() => this.editFunction(x._id)} />
            </div>
            <CardBody>
              <h1 className='fs-3 fw-1 text-
center'>{x.productName}</h1>
              <div className='d-flex justify-
content-between'>
                <h5>Price={x.price}Rs</h5>
                <h5>Category={x.category}</h5>
              </div>
            <div>

```

```

                <h4>Stock={x.inStock}</h4>
            </div>
        </CardBody>
    </Card>
</Col>
)
)}
</Row>
</div>
</div>
)
}
}
export default ProductApi

```

Routing

```

import 'bootstrap/dist/css/bootstrap.min.css';
import './App.css';
import Home from './components/pages/Home';
import About from './components/pages/About';
import Contact from './components/pages/Contact';
import Service from './components/pages/Service';
import { BrowserRouter, Route, Routes } from 'react-router-dom';

/* To use routing we have to install react-router-dom by npm i react-router-dom
then use BrowserRouter at that component in which we call that component which we
want to route
then structure like below '/' when we run project that time this page open
'/about' means if we write localhost:3000/about then we can see only about page
like this..
if we want to change path by click then we have two method
1=> anchor tag
2=> Link component

1=> if we use anchor tag then it is refresh page during click therefore we have
to use link
    example <a href='/about' />
2=> by Link component it is not refresh page
<Link to={'/'} target='_blank'>Home</Link>
target blank open page in new tab

```

```
if we want to send parameter during page open then we have to make nested path  
and during call we can pass anything  
we can pass name . by using this we can make dynamic page like..  
wecome mehul waghela on page we want to show then we have pass that name during  
click  
<Link to={'/Service/1234'}>Service</Link>
```

and we can access parameter by react-router-dom hook useParams

```
let params = useParams()  
  console.log(params)  
  return (  
    <div>  
      Service  
      {params.number}  
    </div>  
  )  
  /* by this page become totally dynamic */  
function App() {  
  return (  
    <div className="">  
      <BrowserRouter>  
        <Routes>  
          <Route path='/' element={<Home />} />  
          <Route path='/About' element={<About />} />  
          <Route path='/Contact' element={<Contact />} />  
          <Route path='/Service' element={<Service />} >  
            <Route path=':number' />  
          </Route>  
        </Routes>  
      </BrowserRouter>  
    </div>  
  );  
}  
  
export default App;
```

Nested Routing

```
import 'bootstrap/dist/css/bootstrap.min.css'  
import { BrowserRouter, Navigate, Route, Routes } from 'react-router-dom';
```

```

import './App.css';
import Home from './components/Pages/Home';
import About from './components/Pages/About';
import Contact from './components/Pages/Contact';
import Contact1 from './components/Pages/NestedPages/Contact1';
import Contact2 from './components/Pages/NestedPages/Contact2';
import Page404 from './components/Page404';
import DummyContact from './components/Pages/DummyContact';

/* in nested routing main point is we dont have to use '/'
for nested routing make route in route and give path 'contact1' like this
to open contact1 page
localhost:3000/contact/contact1 .. we can access this neted page like this
but in this contact and contact1 both page show thots why we have to make Contact
page blank
and in nested path '' blank and that place we use original contact page and below
them another page make therefore only one page show that time.
But for nested routing we have to use >Outlet/> in parent component
if we not use <Outlet> in parent component then nested page doesnot show
and that nested component show according to postion of outlet we use in parent
component if we make parent component first and bottom side Outlet use then then
child component show bottom side of parent component
thots why we have to use <Outlet> only in parent component and make original
component in nested*/
/* Navigate use for path change during starting of page
in this example we write path'/' and in that we navigate to page about then it
take to about page during starting of project but we have that path like below we
have make path of '/about' then we can naviagate*/
/* '*' it means universal path if we write any wrong path then we redirect to
that page we write on universal path
in this exaample we have make 404 page */
function App() {
  return (
    <div className="">
      <BrowserRouter>
        <Routes>
          <Route path='/' element={<Home />} />
          {/* <Route path='/' element={<Navigate to='/about' />} /> */}
          <Route path='/about' element={<About />} />
          <Route path='/contact' element={<DummyContact />} />
          <Route path='' element={<Contact />} />
          <Route path='contact1' element={<Contact1 />} />
          <Route path='contact2' element={<Contact2 />} />
          <Route path='*' element={<Page404 />} />
        </Routes>
      </BrowserRouter>
    </div>
  );
}

```

```

        </Route>
        <Route path='*' element={<Page404 />} />
    </Routes>
</BrowserRouter>
</div>
);
}

export default App;

```

Sidebar.jsx

```

import { Fragment } from "react"
import { Link } from "react-router-dom"

const Sidebar = (Component) => {
    const NewComponent = () => {
        return (
            <Fragment>
                <div className="container-fluid g-0">
                    <div className="row">
                        <div className="col-2 d-flex flex-column px-3" style={{
backgroundColor: '#499195', color: '#fff', height: '100vh' }}>
                            <div className="pb-5"><Link to='/'>HOME</Link></div>
                            <div className="pb-5"><Link
to='/about'>About</Link></div>
                            <div className="pb-5"><Link
to='/contact'>Contact</Link></div>
                            </div>
                            <div className="col-10 g-0">
                                <div className="d-flex justify-content-center"
style={{ backgroundColor: '#70B0E0' }}>
                                    <div className="p-5"><Link
to='/'>HOME</Link></div>
                                    <div className="p-5"><Link
to='/about'>About</Link></div>
                                    <div className="p-5"><Link
to='/contact'>Contact</Link></div>
                                    </div>
                                    <div className="bg-danger-subtle">
                                        <Component />
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </Fragment>
        );
    }
}
```

```

        </div>
      </div>
    </div>
  </Fragment>
)
}
return NewComponent
}
export default Sidebar

```

Page 404

```

import React from 'react'
import { useNavigate } from 'react-router-dom'

const Page404 = () => {
  const navigate = useNavigate()
  const Homepage = () => {
    // But above two method refresh the page that's why navigate method use
    // window.location.reload()/* to reload current page */
    // window.location.href = '/'/* to go to home page or any path */

    /* navigate use. in that method page not refresh */
    navigate('/contact') /* to go to home page or any path */
  }
  return (
    <div className='d-flex justify-content-center align-items-center vh-100'>
      <div className='border border-1 p-5 border-danger border-3 rounded-2
text-center'>
        <h1 className='bg-danger text-white'>Error-404</h1>
        <h2>Page not Found</h2>
        <button onClick={Homepage} className='btn btn-dark mt-3'>Go To
Home Page</button>
      </div>
    </div>
  )
}
export default Page404

```

Dummycontact.jsx

```
import React from 'react'
import Sidebar from '../Sidebar'
import { Outlet } from 'react-router-dom'

/* in this we only use outlet that why we can use nested path and
in nested path we have use main contact page '' like black path */
function DummyContact() {
    return (
        <div>
            <Outlet />
        </div>
    )
}

export default Sidebar(DummyContact)
```

Folder Structure

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - NESTED-ROUTING
 - node_modules
 - public
 - SRC
 - components
 - Pages
 - NestedPages
 - Contact1.jsx
 - Contact2.jsx
 - About.jsx
 - Contact.jsx
 - DummyContact.jsx
 - Home.jsx
 - Page404.jsx
 - Sidebar.jsx
 - App.css
 - App.js
- EDITOR:** Displays the content of `DummyContact.jsx`. The file imports `React` and `Sidebar`, and defines a function `DummyContact` that returns a `<div>` element containing an `<Outlet />`.
- OUTPUT:** Shows a warning from ESLint: "Line 2:25: 'Navigate' is defined but never used `no-unused-vars`". It also indicates that webpack compiled with 1 warning.
- STATUS BAR:** Shows the current line (Ln 12, Col 22), spaces (4), encoding (UTF-8), and file type (JavaScript JSX). It also shows the date and time (8/8/2023, 7:41 PM).

Zoom image api

```
import React, { useEffect, useRef, useState } from 'react'
import axios from 'axios'
import { Col, Container, Form, FormGroup, Input, Label, Row, Spinner, Table } from 'reactstrap'
import { closeImage, openImage } from '../Zoom image/zoomimage'
import Swal from 'sweetalert2'
const ZoomImage = () => {
  let [arr, setarr] = useState([])
  let [obj, setobj] = useState({ hobbies: '' })
  let reference = useRef()
  const setData = () => {
    let formdata = new FormData()
    formdata.append('userImage', obj.userImage)
    formdata.append('firstName', obj.firstName)
    formdata.append('lastName', obj.lastName)
    formdata.append('age', obj.age)
    formdata.append('city', obj.city)
    formdata.append('gender', obj.gender)
    formdata.append('hobbies', obj.hobbies)
    for (let x of formdata.entries()) {
      console.log(x)
    }
    axios.post('https://student-api.mycodelibraries.com/api/user/add',
    formdata)
      .then((res) => {
        console.log(res)
        getData()
      })
      .catch((err) => console.log(err))
  }
  const getData = () => {
    axios.get('https://student-api.mycodelibraries.com/api/user/get')
      .then((res) => {
        arr = res.data.data
        setarr([...arr])
      })
      .catch((err) => console.log(err))
  }
  const deleteapi = (id) => {
```

```
const swalWithBootstrapButtons = Swal.mixin({
  customClass: {
    confirmButton: 'btn btn-success',
    cancelButton: 'btn btn-danger'
  },
  buttonsStyling: false
})

swalWithBootstrapButtons.fire({
  title: 'Are you sure?',
  text: "You won't be able to revert this!",
  icon: 'warning',
  showCancelButton: true,
  confirmButtonText: 'Yes, delete it!',
  cancelButtonText: 'No, cancel!',
  reverseButtons: true
}).then((result) => {
  if (result.isConfirmed) {
    axios.delete(`https://student-
api.mycodelibraries.com/api/user/delete?id=${id}`).then((res) => {
      getData()
    }).catch((err) => {
      console.log(err)
    })
    swalWithBootstrapButtons.fire(
      'Deleted!',
      'Your file has been deleted.',
      'success'
    )
  }
  if (
    /* Read more about handling dismissals below */
    result.dismiss === Swal.DismissReason.cancel
  ) {
    swalWithBootstrapButtons.fire(
      'Cancelled',
      'Your imaginary file is safe :)',
      'error'
    )
  }
})
})
```

```

const editFunction = (id) => {
    axios.get("https://student-api.mycodelibraries.com/api/user/get-user-by-
id?id=" + id).then((res) => {
        obj = res.data.data
        obj.hobbies = obj.hobbies.split(",")
        setobj({ ...obj })
    }).catch((err) => { console.log(err) })
}

const updateapi = () => {
    obj.id = obj._id
    axios.post('https://student-api.mycodelibraries.com/api/user/update',
obj).then((res) => {
        getData()
    }).catch((err) => console.log(err))
}

useEffect(() => {
    // Any website source code we can get like this
    axios.get('https://www.goibibo.com/').then((res) => {
        let html = new DOMParser().parseFromString(res.data, 'text/html')
        console.log(html.getElementsByTagName('title')[0].innerText)
    })
    getData()
}, [])

const changeData = (e) => {
    if (e.target.name === "hobbies") {
        if (e.target.checked) {
            obj.hobbies = [...obj.hobbies, e.target.value]
        }
        else {
            obj.hobbies = obj.hobbies.filter((x) =>
!x.includes(e.target.value))
        }
    }
    else if (e.target.name === 'userImage') {
        obj[e.target.name] = e.target.files[0]
    }
    else {
        obj[e.target.name] = e.target.value
    }
    setobj({ ...obj })
}

```

```

const submitFunction = (e) => {
  e.preventDefault();
  if (obj._id === undefined) {
    setData()
  }
  else {
    updateapi()
  }
  obj = { hobbies: '' }
  setobj({ ...obj })
  reference.current.value = ''
}
return (
  <div>
    <Row>
      <Col xs={6} className="offset-3">
        <Container className="mt-1 py-1 px-4 border border-1 border-
black rounded-2 shadow-lg">
          <h1 className="text-center py-3">Student Form</h1>
          <Form onSubmit={(e) => { submitFunction(e) }}>
            <Row>
              <Col md={6}>
                <FormGroup>
                  <Label for="firstName" className="fw-
600">
                    First Name
                  </Label>
                  <Input
                    id="firstName"
                    name="firstName"
                    placeholder=""
                    type="text"
                    className="main"
                    onChange={changeData}
                    value={obj.firstName || ''}
                  />
                </FormGroup>
              </Col>
              <Col md={6}>
                <FormGroup>
                  <Label for="lastName" className="fw-600
">
                    last Name
                  </Label>
                  <Input

```

```
        id="lastName"
        name="lastName"
        placeholder=""
        type="text"
        className="main"
        onChange={changeData}
        value={obj.lastName || ''}

    />
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
<Label for="age" className="fw-600 ">
    Age
</Label>
<Input
    id="age"
    name="age"
    placeholder=""
    type="number"
    className="main"
    onChange={changeData}
    value={obj.age || ''}
/>
</FormGroup>
</Col>
<Col md={6}>
<FormGroup>
<Label for="city" className="fw-600 ">
    City
</Label>
<select onChange={changeData}
value={obj.city || ''} name="city" className="form-select">
    <option value="surat">Surat</option>
    <option
value="bharuch">Bharuch</option>
    <option
value="vadodara">Vadodara</option>
    </select>
</FormGroup>
</Col>
<Col md={6}>
<Label for="example" className="fw-600 ">
    Gender
</Label>
<Select
    id="example"
    name="example"
    value={obj.gender || ''}
    onChange={changeData}
/>
</Col>
</Row>
</Form>

```

```
</Label>
<div className="d-flex">
  <div>
    <Input
      id="exampleCheck3"
      name="gender"
      type="radio"
      className="gender me-2"
      onChange={changeData}
      value="Male"
      checked={obj.gender === "Male" ||
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"
      onChange={changeData}
      value="Female"
      checked={obj.gender === "Female" ||
obj.gender === "female"}>
    />
    <Label
      check
      for="radio"
      className="px-2">
      Female
    </Label>
  </div>
</div>
<Col md={12} className="">
  <Label
    check
```

```
        for="example"
        className="fw-600
        my-2"
      >
        hobbies
      </Label>
      <Row className="">
        <Col xs={3}>
          <Input
            id="Travelling"
            name="hobbies"
            type="checkbox"
            className="language me-2"
            onChange={changeData}
            value="Travelling"
            checked={obj.hobbies?.includes('T
ravelling')}>
        />
        <Label
          check
          for="Travelling"
          className="px-2"
        >
          Travelling
        </Label>
      </Col>
      <Col xs={3}>
        <Input
          id="Reading"
          name="hobbies"
          type="checkbox"
          className="language me-2"
          onChange={changeData}
          value="Reading"
          checked={obj.hobbies?.includes('R
eading')}>
        />
        <Label
          check
          for="Reading"
          className="px-2"
        >
          Reading
        </Label>
      </Col>
```

```
<Col xs={3}>
  <Input
    id="Exersice"
    name="hobbies"
    type="checkbox"
    className="language me-2"
    onChange={changeData}
    value="Exersice"
    checked={obj.hobbies?.includes('E
xersice')}>
  />
  <Label
    check
    for="Exersice"
    className="px-2">
    Exersice
  </Label>
</Col>
</Row>
<Row className='py-2'>
  <Col xs={12}>
    <Label
      check
      for="Exersice"
      className="py-2">
      Profile
    </Label>
    <input
      id="userImage"
      name="userImage"
      type="file"
      className="language me-2 form-
control"
      onChange={changeData}
      ref={reference}>
    />
  </Col>
</Row>
</Col>
</Row>
<div className="text-center">
  <button className="my-2 btn btn-secondary submit
fs-4">
```

```

                Submit
            </button>
        </div>
    </Form>
</Container>
</Col>
</Row>
<div className="container bg-body-secondary mt-3">
    <h2 className='text-center py-3'>Form</h2>
    <Table className="">
        <thead>
            <tr>
                <th>Sr No</th>
                <th>Profile</th>
                <th>FirstName</th>
                <th>LastName</th>
                <th>Age</th>
                <th>City</th>
                <th>Gender</th>
                <th>Hobbies</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            {arr?.map((x, i) => {
                return <tr key={i + 1}>
                    <td>{i + 1}</td>
                    <td>
                        <img src={x.image} alt="" width={40}
height={40} onClick={() => openImage(x.image)} />
                    </td>
                    <td>{x.firstName}</td>
                    <td>{x.lastName}</td>
                    <td>{x.age}</td>
                    <td>{x.city}</td>
                    <td>{x.gender}</td>
                    <td>{x.hobbies}</td>
                    <td>
                        <button onClick={() => deleteapi(x._id)}
className='me-2 btn text-bg-danger'>Delete</button>
                        <button onClick={() => editFunction(x._id)}
className='btn text-bg-warning'>Edit</button>
                    </td>
                </tr>
            })}
        </tbody>
    </Table>
</div>

```

```

        </tbody>
    </Table>
</div>
<div className="zoom_image">
    <button className='btn btn-close' onClick={closeImage}></button>
    <div className='bg-image'></div>
</div>
</div>
)
}

export default ZoomImage

```

Zoom image function component

```

export const openImage = (img) => {
    document.documentElement.style.setProperty('--display', 'flex')
    document.documentElement.style.setProperty('--image', `url(${img})`)
}
export const closeImage = () => {
    document.documentElement.style.setProperty('--display', 'none')
}

```

Css File

```

.App {
    text-align: center;
}

:root {
    --image: url(assets/images/1200x1200.gif);
    --display: none;
}

body {
    background-color: rgba(81, 111, 170, 0.365);
}

```

```
.bg-image {  
    height: 300px;  
    width: 300px;  
    background-image: var(--image);  
    background-size: cover;  
    background-repeat: no-repeat;  
    animation-name: mehul;  
    animation-duration: 1s;  
}  
  
.btn-close {  
    position: absolute;  
    top: 20px;  
    right: 30px;  
    background-color: #f713139a;  
}  
  
@keyframes mehul {  
    0% {  
        scale: 1;  
    }  
  
    50% {  
        scale: 1.3;  
    }  
  
    100% {  
        scale: 1.6;  
    }  
}
```

ME

Folder structure

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "PRODUCT-MAIN". The "components" folder contains "function" and "Zoom image". "Zoom image" contains "zoomimage.js".
- Editor:** The active file is "zoomimage.jsx". The code implements a "ZoomImage" component using React hooks like useState and useRef.
- Output Panel:** Shows a warning from ESLint: "Line 3:62: 'Spinner' is defined but never used" and "Line 108:8: React Hook useEffect has a missing dependency: 'getData'. Either include it or remove the dependency array".
- System Tray:** Shows the date and time as 14-08-2023.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "PRODUCT-MAIN". The "components" folder contains "Zoom image". "Zoom image" contains "zoomimage.js".
- Editor:** The active file is "zoomimage.js". It exports two functions: "openImage" and "closeImage".
- Output Panel:** Shows a warning from ESLint: "Line 3:62: 'Spinner' is defined but never used" and "Line 108:8: React Hook useEffect has a missing dependency: 'getData'. Either include it or remove the dependency array".
- System Tray:** Shows the date and time as 14-08-2023.

Cookies

```
import React from 'react'
import { useCookies } from 'react-cookie'
const CookiesComponent = () => {
    /* to use cookies in react we have to install react-cookies by this command=>
npm i react-cookie
    then useCookies is use which we destructure in 3 terms
    1=>any name by which we can access cookies data in object form
    2=>set method by which we can set cookies which has path and expiry date also
        path and expiry date is optional
        path is use when we want particular data on particular page for that
requirement we use path
        expiry date is we can set directly or according to current time in this
example i have set time by current time + 2 minute
        after two minutes this cookie is expire
    3=>remove cookie by which we can remove cookie */
    /* */
    let [cookies, setcookies, removecookies] = useCookies()
    const setData = () => {
        let expiryDate = new Date()
        expiryDate.setMinutes(expiryDate.getMinutes() + 2)
        setcookies('greet', 'Mehul Waghela', { path: '/', expires: expiryDate })
        setcookies('greetNew', 'Sagar Davara', { path: '/home' })
    }
    const getData = () => {
        console.log(cookies)
    }
    const removeData = () => {
        removecookies('greet')
    }
    return (
        <div>
            <div>
                {cookies.greet}
            </div>
            <div>
                <button onClick={setData}>Set Cookies</button>
                <button onClick={getData}>Get Cookies</button>
                <button onClick={removeData}>Remove Cookies</button>
            </div>
        </div>
    )
}
```

```
export default CookiesComponent
```

Reducer Component

```
import React, { useReducer } from 'react'
import { myReducer } from './Reducer'

/* in reducer we have state and dispatch available.
State means any data which we store and dispatch is method which we call to take
particular work from useReducer.
In state we can store any data and use reducer has two parameter one is reducer
function name and 2 nd parameter is data which we store in state.
to use reducer call dispatch method with any data. reducer function do work
according to data

in regular practice state is array and we pass from dispatch it is object in
which first key is type : which work we want to do from useReducer that is set in
object type key and second key is any data which we have to pass do that
particular work
and in useReducer function we regular use switch case according to data pass
switch case check work and data and do that work and return state*/
const ReducerComponent = () => {
  let [state, dispatch] = useReducer(myReducer, 'hello')
  return (
    <div>
      <h1>{state}</h1>
      <button onClick={() => dispatch('Mehul')}>Increment Task</button>
    </div>
  )
}

export default ReducerComponent
```

Reducer.js

```
/* reducer is one type of function in which two parameter occur one is state and
another is action. in action we get data on which we work and save in state and
return that state where we call the reducer */

export const myReducer = (state, action) => {
```

```
    state = state + " " + action
    return state
}
```

Redux first-app

App.js

```
import './App.css';
import Component1 from './components/Component1';
import Component2 from './components/Component2';

/* for make simple redux file first we have to install 3 files by this command
=> npm i redux react-redux redux-thunk
then we have to make four folder
1=>reducers
2=>store
3=>type
4=>action

1=>reducer folder we have all reducer and one file is root reducer in which we
combine multiple reducer in single object by using a combineReducers method import
from redux
1=> 1=>counter reducer is make in this reducer we make default state any type
object , array or any data type and it is set in state and if our defaultstate is
object then we have to return only object
and this state work same as normal reducer

and during dispatch method call then compiler find that type in root reducer and
in which reducer find that type then that process run therefore we have to
require different type in all reducers thats why we make type.js file in which we
make cont type and we use at all place where we use that type. due to const we
not use same type another time

2=>Store.js file in which we use method createStore in which we use root reducer
and applyMiddleware we use thunk or saga which is middleware which is use to
access data in all components

3=>Type.js in this file we make unique type therefor no any conflict occur
```

4=>Action.js in this file we make functions to call all dispatch method dynamic from any component we require data then we have to call function and we can access any data directly from any component by function call

then we have to use Provider and use attribute store use which name is we can not change another all name we can change but store name attribute we can not change

in this store we send argument of store which we created

then we can use that store by useSelector and we can use dispatch from "useDispatch"

```
/*
function App() {
  return (
    <div>
      <Component1 />
      <Component2 />
    </div>
  );
}

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { Provider } from 'react-redux';
import { store } from './redux/store/store';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}>
    <App />
  </Provider>
);
reportWebVitals();
```

Action.js

```
import { DECREMENT, INCREMENT } from "../type/type"

export const incrementCount = (number) => {
    return { type: INCREMENT, number: number }
}

export const decrementCount = (number) => {
    return { type: DECREMENT, number: number }
}
```

Counter reducer

```
import { DECREMENT, INCREMENT } from "../type/type"

let defaultState = {
    count: 0
}

export const counterReducer = (state = defaultState, action) => {
    console.log(state)
    console.log(action)
    switch (action.type) {
        case INCREMENT: {
            return {
                count: state.count + action.number
            }
        }
        case DECREMENT: {
            return {
                count: state.count - action.number
            }
        }
        default: {
            return defaultState
        }
    }
}
```

Root reducer

```
import { combineReducers } from "redux";
import { counterReducer } from "./counterReducer";

export const rootReducer = combineReducers({
  counter: counterReducer,
  number: counterReducer
})
```

Store reducer

```
import { applyMiddleware, createStore } from "redux";
import { rootReducer } from "../reducers/rootReducer";
import thunk from "redux-thunk";

export const store = createStore(rootReducer, applyMiddleware(thunk))
```

type.js

```
export const INCREMENT = "INCREMENT"
export const DECREMENT = "DECREMENT"
```

Component1.jsx

```
import React from 'react'
import { useDispatch, useSelector } from 'react-redux'
import { decrementCount, incrementCount } from '../redux/action/action'

const Component1 = () => {
  const state = useSelector((state) => state)
  const dispatch = useDispatch()
  console.log(state)
  return (
    <>
      <div>Component1</div>
      <h1>{state.counter.count}</h1>
    </>
  )
}
```

```
        <button onClick={() =>
dispatch(incrementCount(5))}>Increment</button>
        <button onClick={() =>
dispatch(decrementCount(10))}>Decrement</button>
    </>
)
}

export default Component1
```

Component2.jsx

```
import React from 'react'
import { useSelector } from 'react-redux'

const Component2 = () => {
    let state = useSelector((state) => state)
    return (
        <div>
            <h1>{state.counter.count}</h1>
            <div>Component2</div>
        </div>
    )
}

export default Component2
```

Folder structure

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "REDUX-PRACTICE". The "type" folder is selected.
- Code Editor:** Displays the content of "Component2.jsx".
- Terminal:** Shows the command line output and current working directory.
- Bottom Bar:** Includes the Windows taskbar with various icons.

```
src > components > Component2.jsx > ...
1 import React from 'react'
2 import { useSelector } from 'react-redux'
3
4 const Component2 = () => {
5     let state = useSelector((state) => state)
6     return (
7         <div>
8             <h1>{state.counter.count}</h1>
9             <div>Component2</div>
10        </div>
11    )
12 }
13
14 export default Component2
15
```

Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.
D:\@Mehul Waghela\react\Task\redux-practice>