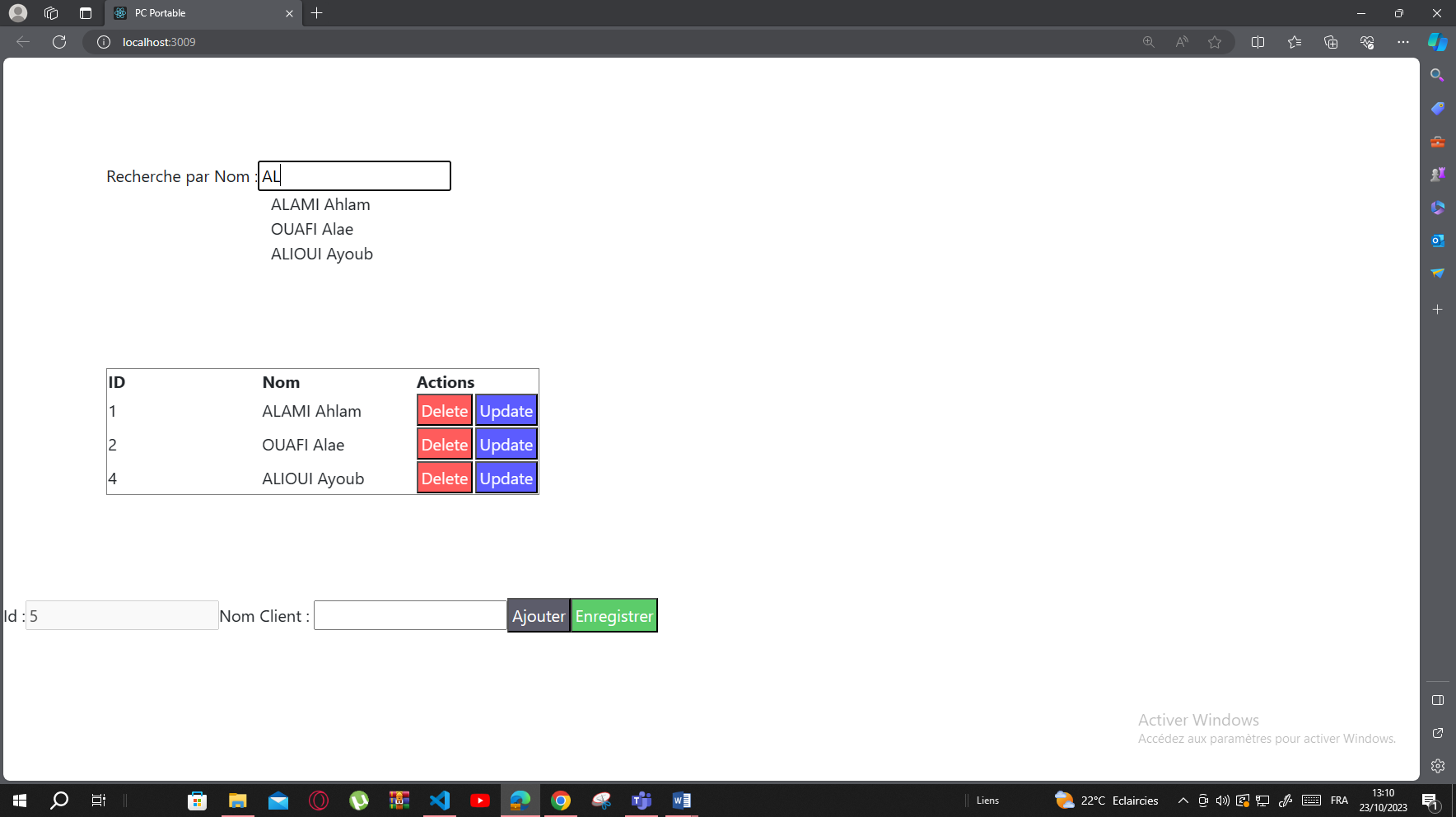
FRONT END FULL STACK 201 - 202 -205 -206 IFMOTICA FES

***REACT TP 06* composants fonction et class props et state**

**Créer le composant class Client.js suivant**



**Client.js**

import React from "react";

export default class Client  extends React.Component{

    Tc=[{id:1,Nom:"ALAMI Ahlam"},{id:2,Nom:"OUAFI Alae"},{id:3,Nom:"TAZI OUAFAE"},

 {id:4,Nom:"ALIOUI Ayoub"}]

 state={TClient:*this*.Tc,id:5,NomAj:"",txtRech:""}

 supprimer=(idsup)=>{

*this*.setState({TClient:[...*this*.state.TClient.filter((cli)=>{return cli.id!==idsup })]})

 }

 ajouter=(e)=>{

    const T=[...*this*.state.TClient];

    T.push({id:*this*.state.id,Nom:*this*.state.NomAj})

*this*.setState({id:*this*.state.id+1,NomAj:""})

*this*.setState({TClient:T})

 }

 afficher=(idaffi)=>{

    const client=*this*.state.TClient.find((cli)=>cli.id===idaffi)

*this*.setState({id:client.id,NomAj:client.Nom})

 }

 enregistrer=()=>{

     const possup=*this*.state.TClient.findIndex((cli)=>cli.id===*this*.state.id)

     const T=[...*this*.state.TClient];

     T.splice(possup,1,{id:*this*.state.id,Nom:*this*.state.NomAj})

*this*.setState({TClient:T})

 }

 rechercher=()=>{

const Trech=*this*.Tc.filter((cli)=> cli.Nom.toLowerCase().search(document.getElementById("idRech").value.toLowerCase())!==-1)

*this*.setState({TClient:Trech})

 }

 render(){

    return <>

    <div style={{margin:"100px"}}>

 Recherche par Nom :<input type="text" id="idRech"  onChange={()=>{*this*.rechercher()}}/>

 {*this*.state.TClient.map((cli)=>{ return <div style={{marginLeft:"160px"}}>{cli.Nom}</div>})}

 </div> <table border="1"  style={{margin:"100px"}}>

        <tr><th>ID</th><th>Nom</th><th colSpan={2}>Actions</th></tr>

        { *this*.state.TClient.map((cli)=>{ return <tr key={cli.id}><td width="150px" >{cli.id} </td><td width="150px" >{cli.Nom} </td>

        <td><button style={{color:"#fff",padding:"3px", backgroundColor:"#FF5C5C" }} onClick={()=>{*this*.supprimer(cli.id)}}>Delete</button></td>

        <td><button style={{color:"#fff",padding:"3px", backgroundColor:"#5C5CFF" }} onClick={()=>{*this*.afficher(cli.id)}}>Update</button></td>

        </tr>})}

    </table>

    Id  :<input type="text" value={*this*.state.id} disabled ></input>

    Nom Client : <input type="text" id="idtn" value={*this*.state.NomAj} onChange={(e)=>{*this*.setState({NomAj:document.getElementById("idtn").value})}} ></input>

    <button style={{color:"#fff",padding:"3px", backgroundColor:"#5C5C6a" }} onClick={()=>{*this*.ajouter()}}>Ajouter</button>

    <button style={{color:"#fff",padding:"3px", backgroundColor:"#5CcC6a" }} onClick={()=>{*this*.enregistrer()}}>Enregistrer</button>

    </>

 }

}

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import Client from './Client';

const root1 = ReactDOM.createRoot(document.getElementById('root'));

root1.render(<Client></Client>)

*// If you want to start measuring performance in your app, pass a function*

*// to log results (for example: reportWebVitals(console.log))*

*// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals*

**Index.css**

body {

  margin: 0;

  font-family: *-apple-system*, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

*sans-serif*;

  -webkit-font-smoothing: *antialiased*;

  -moz-osx-font-smoothing: *grayscale*;

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

*monospace*;

}

.buttonStyle {

  color: #fff;

  background-color:#ec5f0d;

  }