import { React, Component } from 'react';

class Student extends Component {

constructor(){

super();

this.state = {

act : 0,

idx : '',

datas : []

}

}

componentDidMount(){

fetch('http://localhost:8080/api/list')

.then(response=>response.json()).then(data=>{

console.log(data)

this.setState({datas:data})

}

);

}

handleSubmit=(e)=>{

e.preventDefault();

let datas = this.state.datas;

let id = this.refs.id.value;

let name = this.refs.name.value;

let specification = this.refs.specification.value;

let dob = this.refs.dob.value;

let createddate = this.refs.createddate.value;

if(this.state.act === 0)

{

let data = {

"id":id,

"name" : name,

"specification" : specification,

"dob":dob,

"createddate":createddate

}

datas.push(data);

fetch(`http://localhost:8080/api/create`,{

method : 'POST',

headers:{

"Content-Type" : "application/json"

},

body : JSON.stringify(data)

}).then(response=>response.json()).then(data=>{

console.log(data)

});

}

else

{

let index = this.state.idx;

datas[index].id = id;

datas[index].name = name;

datas[index].specification = specification;

datas[index].dob = dob;

datas[index].createddate = createddate;

let data = {

"id":id,

"name" : name,

"specification" : specification,

"dob":dob,

"createddate":createddate

}

fetch(`http://localhost:8080/api/update`,{

method : 'PUT',

headers:{

"Content-Type" : "application/json"

},

body : JSON.stringify(data)

}).then(response=>response.json()).then(data=>{

console.log(data)

let updatedStudents=[this.state.student];

this.setState({teacher:updatedStudents});

});

}

this.setState({

datas : datas,

act : 0

})

this.refs.myForm.reset();

this.refs.id.focus();

}

handleDelete = (index,obj) =>{

let datas = this.state.datas;

datas.splice(index,1);

this.setState({

datas:datas

})

this.refs.id.focus();

fetch(`http://localhost:8080/api/delete/${obj.id}`,{

method:'DELETE',

}) .then(response=>response.json()).then(data=>{

console.log(data)

});

}

async remove(id)

{

console.log(id)

let updatedStudents=[this.state.student].filter(i=>i.id!==id);

this.setState({student:updatedStudents});

}

handleEdit = (index) => {

let data = this.state.datas[index];

this.refs.id.value = data.id;

this.refs.name.value = data.name;

this.refs.specification.value = data.specification;

this.refs.dob.value = data.dob;

this.refs.createddate.value = data.createddate;

this.setState({

act: 1,

idx : index

})

}

async update()

{

// fetch(`/api/update`,{

// method:'PUT',

// headers:{

// 'Accept':'application/json',

// 'Content-Type':'application/json'

// }

// }).then(()=>{

// let updatedStudents=[this.state.student];

// this.setState({student:updatedStudents});

// });

}

render() {

var curr = new Date();

curr.setDate(curr.getDate());

var createddate = curr.toISOString().substr(0,10);

let datas = this.state.datas;

return (

<div className="App">

<form ref="myForm" className="myForm">

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

<h2>Student registration</h2>

<label>Enrollment number</label>

<input type="text" ref="id" placeholder="Enter roll number" className="formField"/>

<label>Name</label>

<input type="text" ref="name" placeholder="Enter name" className="formField"/>

<label>Specification</label>

<input type="text" ref="specification" placeholder="Enter specification" className="formField"/>

<label>Dob</label>

<input type="Date" ref="dob" max="2002-01-01" min="1997-01-01" placeholder="Enter dob" className="formField"/>

#for taking the calender between 1997 to 2002 only in this single line it will be done

<label>Created date</label>

<input type="dateRequired" ref="createddate" defaultValue={createddate} name="dateRequired" placeholder="Enter created date" className="formField"/>

#these lines which are in pink will helpful for making the date set to todays date in forms

<button onClick={e => this.handleSubmit(e)} className="myButton"> Save</button>

</form>

<br/>

<br/>

<table className="table">

<tr>

<th>Enrollment number</th>

<th>Name</th>

<th>Specification</th>

<th>Dob</th>

<th>Created date</th>

<th>Delete</th>

<th>Edit</th>

</tr>

{this.state.datas.map((data,index)=>

<tr key={index}>

<td>{data.id}</td>

<td>{data.name}</td>

<td>{data.specification}</td>

<td>{data.dob} </td>

<td>{data.createddate} </td>

<td>

<button onClick={e => this.handleDelete(index,data)} className="myListButton">Delete</button>

</td>

<td>

<button onClick={e => this.handleEdit (index)} className="myListButton">Edit</button>

</td>

</tr>

)

}

</table>

</div>

);

}

}

export default Student;