**firebase.js**

import firebase from 'firebase/compat/app';

import 'firebase/compat/database';

const firebaseConfig = {

apiKey: "AIzaSyBr-MMnTPjp0bwOXVMiY7RIj3Rqduv9BF0",

authDomain: "learn-6babf.firebaseapp.com",

projectId: "learn-6babf",

storageBucket: "learn-6babf.appspot.com",

messagingSenderId: "812576426474",

appId: "1:812576426474:web:d89a3b91b2e1c6cad6ca93"

};

// Initialize Firebase

const firebaseDB = firebase.initializeApp(firebaseConfig);

export default firebaseDB.database().ref();

**Register.js**

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

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

import firebaseDB from './firebase';

const Register = () => {

const history = useNavigate();

const [data, setData] = useState({

firstname: '',

lastname: '',

email: '',

});

const [getData, setGetData] = useState({});

useEffect(() => {

const fetchData = async () => {

try {

const datadetails = await firebaseDB.child('register').get();

if (datadetails.exists()) {

setGetData(datadetails.val());

}

} catch (error) {

console.error('Error fetching data:', error);

}

};

fetchData();

}, []);

const { firstname, lastname, email } = data;

const changeHandler = e => {

setData({ ...data, [e.target.name]: e.target.value });

};

const submitHandler = async e => {

e.preventDefault();

try {

await firebaseDB.child('register').push(data);

setData({

firstname: '',

lastname: '',

email: '',

});

console.log('Data saved successfully!');

} catch (err) {

console.error('Error saving data:', err);

}

};

const deleteHandler = key => {

firebaseDB.child(`register/${key}`).remove(err => {

if (err) {

console.log(err);

}

});

};

return (

<div>

<center>

<h2>Register Form</h2>

<form onSubmit={submitHandler}>

<label>First Name</label><br />

<input type="text" name="firstname" placeholder="First Name" value={firstname} onChange={changeHandler} /><br />

<label>Last Name</label><br />

<input type="text" name="lastname" placeholder="Last Name" value={lastname} onChange={changeHandler} /><br />

<label>Email</label><br />

<input type="text" name="email" placeholder="Email" value={email} onChange={changeHandler} /><br />

<input type="submit" value="Register" />

</form>

</center>

<div>

{Object.keys(getData).map(key => (

<div key={key} style={{ margin: '10px' }}>

First Name: {getData[key].firstname}<br />

Last Name: {getData[key].lastname}<br />

Email: {getData[key].email}<br />

<button onClick={() => history(`/edit/?firstname=${getData[key].firstname}&lastname=${getData[key].lastname}&email=${getData[key].email}&key=${key}`)}>Update</button>

<button onClick={() => deleteHandler(key)}>Delete</button>

</div>

))}

</div>

</div>

);

};

export default Register;

**Edit.js**

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

import { useLocation, useNavigate } from 'react-router-dom';

import firebaseDB from './firebase';

const Edit = () => {

const location = useLocation();

const query = useMemo(() => new URLSearchParams(location.search), [location.search]);

const history = useNavigate();

const [data, setData] = useState({

firstname: '',

lastname: '',

email: '',

});

const { firstname, lastname, email } = data;

const changeHandler = e => {

setData({ ...data, [e.target.name]: e.target.value });

};

useEffect(() => {

setData({

firstname: query.get('firstname'),

lastname: query.get('lastname'),

email: query.get('email'),

});

}, [query]);

const submitHandler = e => {

e.preventDefault();

const key = query.get('key');

if (key) {

firebaseDB.child(`register/${key}`).set(

data,

err => {

if (err) {

console.log(err);

} else {

// Navigate back to the register page without explicitly reloading

history(`/`); // Replace '/register' with your actual route

}

}

);

}

};

return (

<div>

<center>

<h2>Edit Form</h2>

<form onSubmit={submitHandler}>

<label>First Name</label><br />

<input type="text" name="firstname" placeholder="First Name" value={firstname} onChange={changeHandler} /><br />

<label>Last Name</label><br />

<input type="text" name="lastname" placeholder="Last Name" value={lastname} onChange={changeHandler} /><br />

<label>Email</label><br />

<input type="text" name="email" placeholder="Email" value={email} onChange={changeHandler} /><br />

<input type="submit" value="Save" />

</form>

</center>

</div>

);

};

export default Edit;