import { LightningElement } from 'lwc';

export default class ExchangeCurrency extends LightningElement {

    showOutput = false;

    convertedValue = "";

    fromCurrency = "";

    toCurrency = "";

    enteredamount = "";

    currencyOptions = []; //we have to fetch the currency options with the help of API call.

    connectedCallback(){

        this.fetchSymbol();

    }

    changeHanlder(event){

        //here I have to get the inputs first. That is why I have to destructure the elements

        let{name, value} = event.target;

        //now if the name is amount then I have to populate the amount into one of the property named as enteredamount

        if(name === "amount")

            this.enteredamount = value;

        if(name === "fromCurrency")

            this.fromCurrency = value;

        if(name === "toCurrency")

            this.toCurrency = value;

    }

    clickHandler(){

        this.conversion();

    }

    //here we have to do the API call to the server to get the currency symbol. In javascript we have to do the asynchronous API call

    async fetchSymbol(){

        let endpoint = 'https://api.frankurter.app/currencies';

        try{

            let response = await fetch(endpoint);

            if(!response.ok){

                throw new Error('Network response was not ok');

            }

            const data = await response.json()

            let options = [];

            for(let symbol in data){

                options = [...options, {label : symbol, value : symbol}]

            }

            this.currencyOptions = [...options];

        }catch(error){

            console.log(error);

        }

    }

    async conversion(){

        let endpoint = `https://api.frankurter.app/latest?amount=${this.enteredamount}&from=${this.fromCurrency}&to={this.toCurrency}`; //to dynamically access the amount of from currency and the to currency, we have to write the endpoint with the help of the string interpolation tecnique

        try{

            let response = await fetch(endpoint);

            if(!response.ok){

                throw new Error('Network response was not ok');

            }

            const data = await response.json()

            this.convertedValue = data.rates[this.toCurrency];

            this.showOutput = true;

        }catch(error){

            console.log(error);

        }

    }

}