Graphical user interface, website

Description automatically generated

HTML Code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Currency Converter</title>

    <link rel="stylesheet" href="style.css">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

</head>

<body>

    <div class="container">

        <h2 class="heading">Currency Converter</h2>

        <div class="rate">

            <div class="rate1"></div>

            <div class="rate2"></div>

        </div>

        <div class="main">

            <div class="initial">

                <div class="options"><select></select></div>

                <div class="input"><input type="number"></div>

            </div>

            <span class="material-icons swap" style="font-size: 36px;">swap\_horiz</span>

            <div class="target">

                <div class="options"><select></select></div>

                <div class="input"><input disabled type="number"></div>

            </div>

        </div>

        <button class="result">Result</button>

    </div>

    <script src="sketch.js"></script>

</body>

</html>

CSS code:

@import url('https://fonts.googleapis.com/css2?family=Yantramanav:wght@500&display=swap');

body{

    height: 100vh;

    display: flex;

    justify-content: center;

    align-items: center;

    /\* background-image: linear-gradient(to bottom,purple,black); \*/

    background-image: url('back3.jpg');

    background-repeat: no-repeat;

    background-attachment: fixed;

    background-size: 100% 100%;

    overflow: hidden;

}

.container{

    background-color: white;

    width: 470px;

    height: 360px;

    padding: 70px;

    box-shadow: 0px 0px 50px rgba(0, 0, 0, 0.5);

    border-radius: 40px;

    font-family: "Yantramanav",sans-serif;

}

.heading{

    color: #46406c;

    text-align: center;

    text-transform: uppercase;

    font-size: 2.8rem;

}

.rate{

    margin-top: -1.5rem;

}

.rate1{

    color: #848484;

}

.rate2{

    margin-top: 0.5rem;

    font-size: 1.5rem;

}

.main{

    display: flex;

    flex-direction: row;

    justify-content: space-between;

    margin-top: 2rem;

}

.swap{

    align-self: flex-end;

    padding: 5px;

    cursor: pointer;

}

.options select{

    padding: 2%;

    width: 75px;

}

.input input{

    all: unset;

    display: inline-block;

    margin-top: 1rem;

    width: 175px;

    height: 48px;

    padding: 5px;

    background-color: #ebebeb;

    color: #848484;

    font-size: 1.5rem;

    border: none;

    box-sizing: border-box;

    border-radius: 5px;

    font-weight: 500;

    appearance: text;

}

.target{

    transform: translate(-5%);

}

.result{

    margin-top: 1.5rem;

    width: 100%;

    height: 50px;

    text-align: center;

    color: #fff;

    background-color: #46406c;

    outline: none;

    border: none;

    border-radius: 5px;

}

.result:active{

    background-color: #342f52;

}

Js Code:

let rate1 = document.querySelector(".rate1");

let rate2 = document.querySelector(".rate2");

let resultBtn = document.querySelector(".result");

let selects = document.querySelectorAll(".options select");

let sel1 = selects[0];

let sel2 = selects[1];

let inputs = document.querySelectorAll(".input input");

let inpt1 = inputs[0];

let inpt2 = inputs[1];

let rates = {};

let requestURL = "https://api.exchangerate.host/latest?base=USD";

fetchRates();

async function fetchRates(){

    let res = await fetch(requestURL);

    res = await res.json();

    rates = res.rates;

    populateOptions();

}

function populateOptions(){

    let val = "";

    Object.keys(rates).forEach(code=>{

        let str = `<option value="${code}">${code}</option>`;

        val += str;

    })

    selects.forEach((s) => (s.innerHTML = val));

}

function convert(val,fromCurr,toCurr){

    let v = (val/rates[fromCurr])\*rates[toCurr];

    let v1 = v.toFixed(3);

    return v1 = 0.0 ? v.toFixed(5):v1;

}

function displayRate(){

    let v1 = sel1.value;

    let v2 = sel2.value;

    let val = conver(1,v1,v2);

    rate1.innerHTML = `1 ${v1} equals`;

    rate2.innerHTML = `${val} {v2}`;

}

resultBtn.addEventListener("click", ()=>{

    let fromCurr = sel1.value;

    let fromVal = parseFloat(inpt1.value);

    let toCurr = sel2.value;

    if(isNaN(fromVal)){

        alert("Enter a Number");

    }else{

        let cVal =  convert(fromVal,fromCurr,toCurr);

        inpt2.value = cVal;

    }

});

selects.forEach(s=>s.addEventListener("change",displayRate));

document.querySelector(".swap").addEventListener("click",()=>{

    let in1  = inpt1.value;

    let in2 = inpt2.value;

    let op1 = sel1.value;

    let op2 = sel2.value;

    inpt2.value = in1;

    inpt1.value = in2;

    sel2.value = op1;

    sel1.value = op2;

    displayRate();

})