**Working Code :**

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

    <div>

        <input type="text" id="amount">

        <p id = "balance"></p>

        <button id="deposit">Deposit</button>

        <button id="withdraw">Withdraw</button>

    </div>

    <!--<script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js"></script>-->

    <!--<script src="https://cdn.jsdelivr.net/gh/ethereum/web3.js@1.0.0-beta.36/dist/web3.min.js" integrity="sha256-nWBTbvxhJgjslRyuAKJHK+XcZPlCnmIAAMixz6EefVk=" crossorigin="anonymous"></script>-->

    <script src="https://cdn.jsdelivr.net/gh/ethereum/web3.js@1.0.0-beta.36/dist/web3.min.js"></script>

    <!--<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" crossorigin="anonymous"></script>-->

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>

    <script>

        if (typeof web3 !== 'undefined') {

            web3 = new Web3(web3.currentProvider);

        } else {

            // set the provider you want from Web3.providers

            web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8080"));

        }

    </script>

    <script>

        var contract;

        $(document).ready(function() //whenever page is ready create & initialize object of web3

        {

            //web3 = new Web3(web3.currentProvider);  //InjectedWeb3

            //Web3 = require('web3');

            var address = "0x5F22C7dB79Cd197faAb170dd877DF7527cBC828E";

            var abi = [

    {

        "inputs": [

            {

                "internalType": "int256",

                "name": "amt",

                "type": "int256"

            }

        ],

        "name": "deposit",

        "outputs": [],

        "stateMutability": "nonpayable",

        "type": "function"

    },

    {

        "inputs": [],

        "stateMutability": "nonpayable",

        "type": "constructor"

    },

    {

        "inputs": [

            {

                "internalType": "int256",

                "name": "amt",

                "type": "int256"

            }

        ],

        "name": "withdraw",

        "outputs": [],

        "stateMutability": "nonpayable",

        "type": "function"

    },

    {

        "inputs": [],

        "name": "getBalance",

        "outputs": [

            {

                "internalType": "int256",

                "name": "",

                "type": "int256"

            }

        ],

        "stateMutability": "view",

        "type": "function"

    }

];

            contract = new web3.eth.Contract(abi, address);   //ABI , Address

            contract.methods.getBalance().call().then(function(bal)

            {

                $('#balance').html(bal);

            })

        })

        $('#deposit').click(function()

        {

            var amt=0;

            amt=parseInt($('#amount').val());

            web3.eth.getAccounts().then(function(accounts){ //requestAccounts

                var acc=accounts[0];

                return contract.methods.deposit(amt).send({from: acc});

            }).then(function(tx){

                console.log(tx);

            }).catch(function(tx){

                console.log(tx);

            })

        })

        $('#withdraw').click(function()

        {

            var amt=0;

            amt=parseInt($('#amount').val());

            web3.eth.getAccounts().then(function(accounts){

                var acc=accounts[0];

                return contract.methods.withdraw(amt).send({from: acc});

            }).then(function(tx){

                console.log(tx);

            }).catch(function(tx){

                console.log(tx);

            })

        })

    </script>

</body>

</html>

**Contract code:**

//SPDX-License-Identifier: MIT

pragma solidity ^0.8.8;

contract Bank{

    int bal;

    constructor() {

        bal=1;

    }

    function getBalance()public view returns(int){

        return bal;

    }

    function withdraw(int amt) public{

        bal-=amt;

    }

    function deposit(int amt) public{

        bal+=amt;

    }

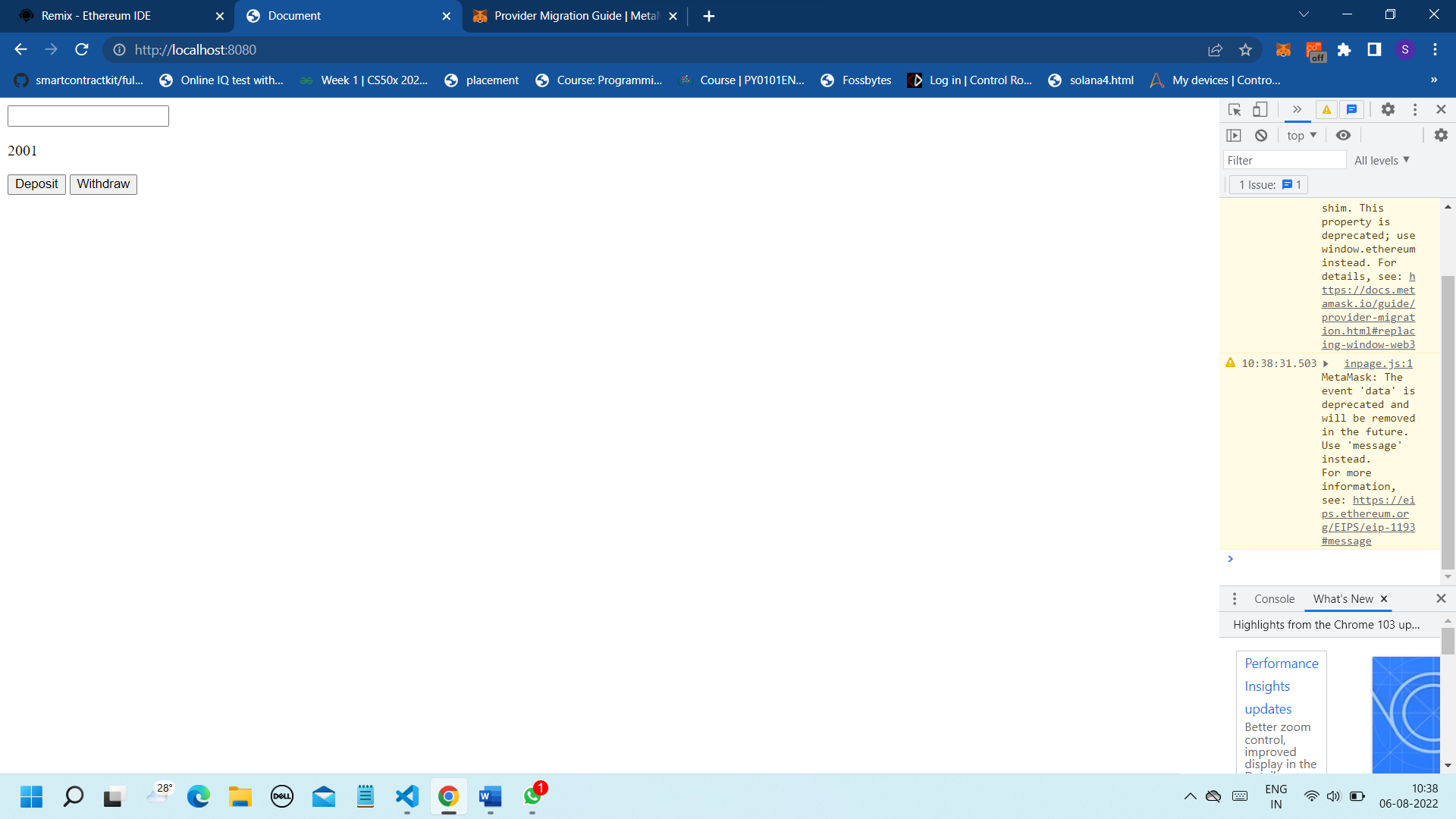
}

**A screenshot of a computer

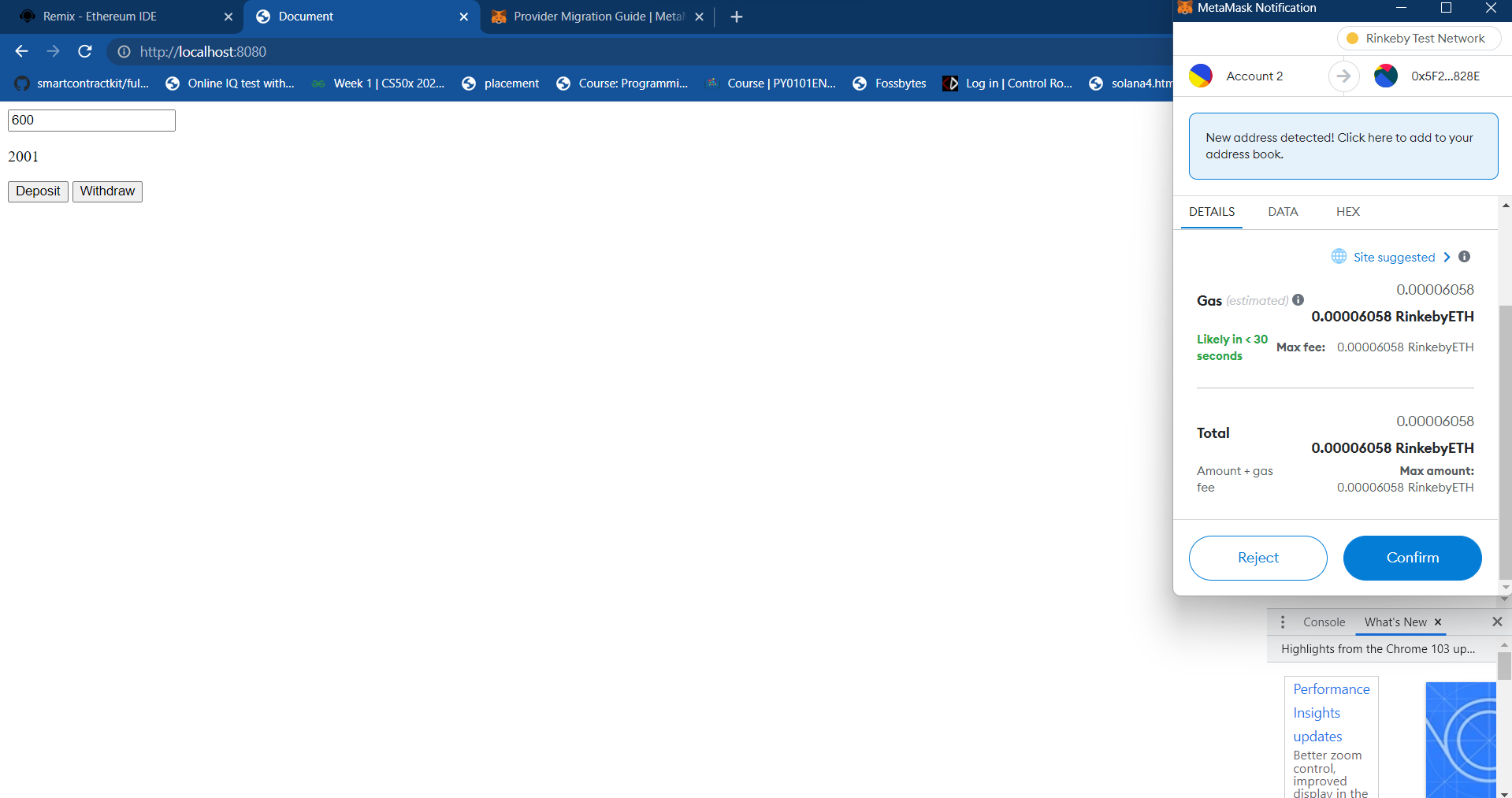
Description automatically generated**



**Displaying balance**

****

**Deposit**

****

**Generated hash**

**Graphical user interface, application, Word

Description automatically generated**

**After successful deposit**

**Graphical user interface, text, application, Word

Description automatically generated**

**Withdrawal – Before**

**Graphical user interface, application, Word

Description automatically generated**

**Hash generated**

**Graphical user interface, text, application, Word

Description automatically generated**

**Withdrawal successful**

**Graphical user interface, text, application, Word

Description automatically generated**