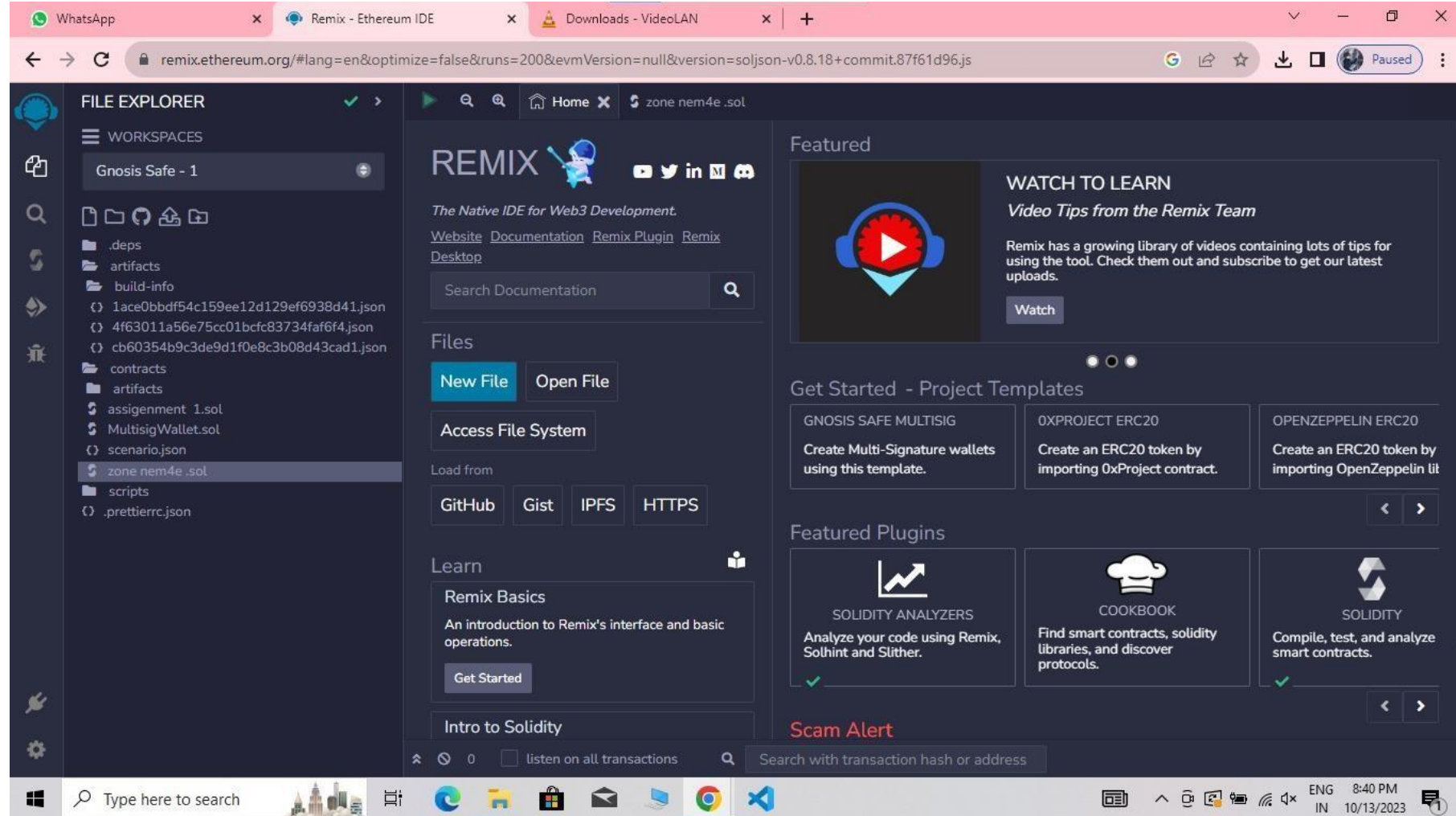


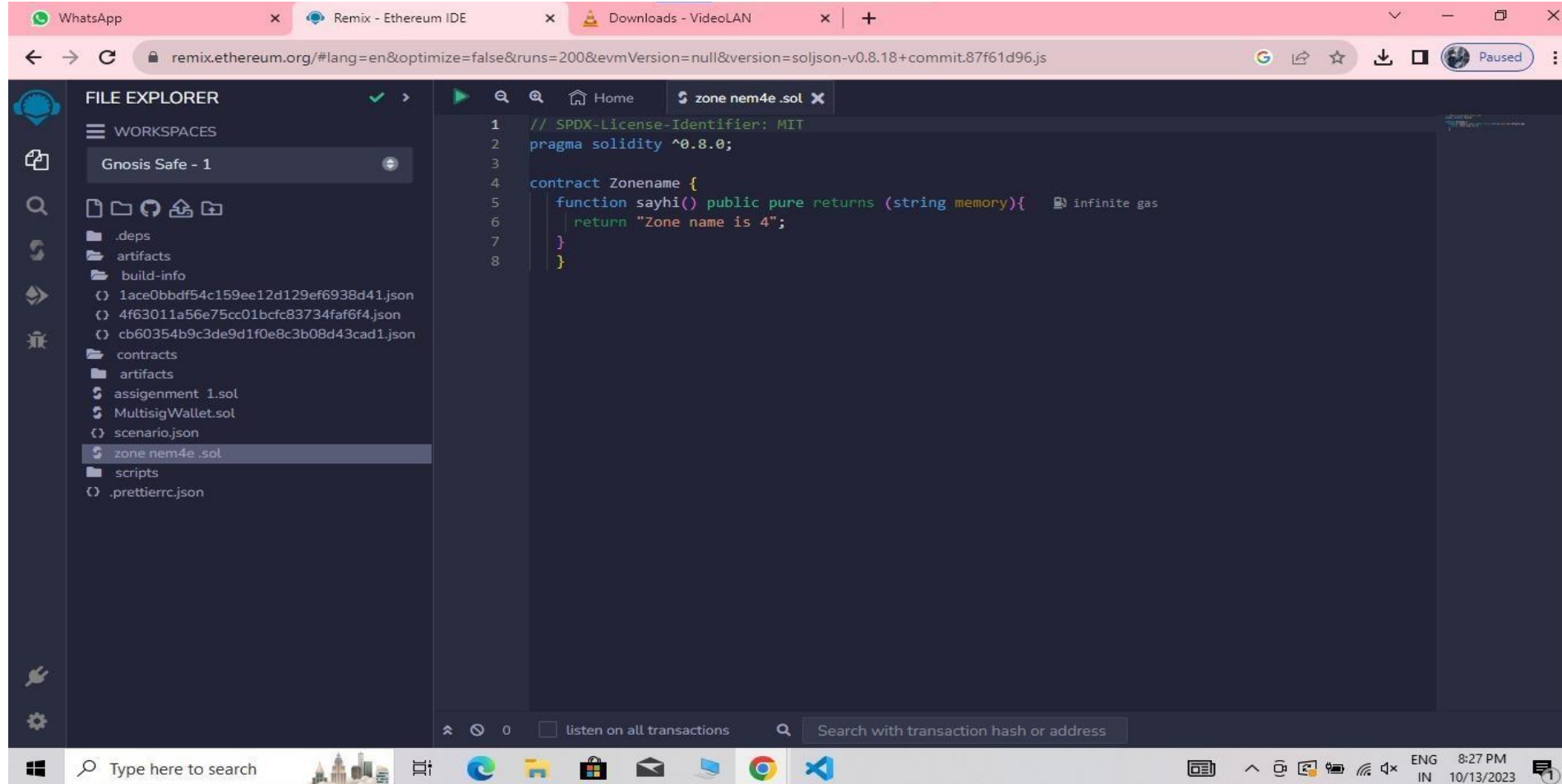
# Assignment 1

Name	Kirubharathi R
ZONE	4
COLLAGE	GKM COLLAGE OF ENGINEERING AND TECHNOLOGY


# 1. GO TO THE CHROME ATFORM OPEN REMIX PLARTF

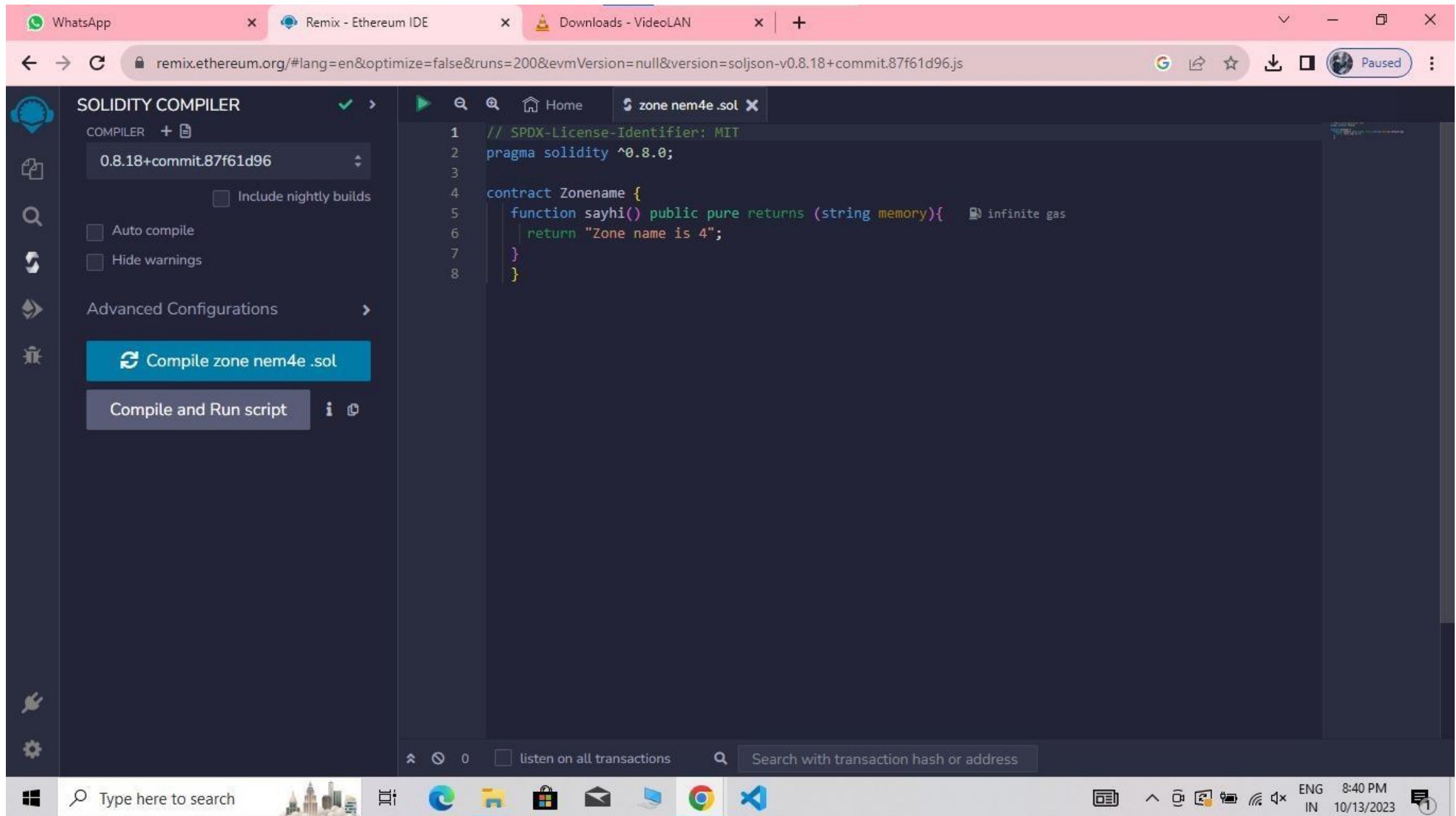


## 2. OPEN THE REMIX PAGE AND CRRATE A NEW FILE



3. IN THE NEWLY CREATED FILE,CREATE A PROGRAM TO RETURN YOUR STRING ,”ZONE NAME”

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract udhaya{
5     function zoon8() public pure returns(string memory){  infinite gas
6         return "zoon 4";
7     }
8 }
```



## 4. SAVE THE PROGRAM AND COMPILE IT TO GET THE ABI AND BYTECODE

```
[
  {
    "inputs": [],
    "name": "zoon8",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "pure",
    "type": "function"
  }
]
```

**ABI**

# BYTECODE

- 608060405234801561001057600080fd5b5061017c806100206000396000f  
3fe608060405234801561001057600080fd5b506004361061002b57600035  
60e01c8063cc5ba50214610030575b600080fd5b61003861004e565b60405  
161004591906100c4565b60405180910390f35b6060604051806040016040  
5280600681526020017f7a6f6f6e20340000000000000000000000000000  
000000000000000000000000000000000815250905090565b6000610096826100e6565  
b6100a081856100f1565b93506100b0818560208601610102565b6100b98  
1610135565b840191505092915050565b600060208201905081810360008  
301526100de818461008b565b905092915050565b6000815190509190505  
65b600082825260208201905092915050565b60005b83811015610120578  
082015181840152602081019050610105565b8381111561012f576000848

401525b50505050565b6000601f19601f830116905091905056fea2646970  
667358221220085089e332d62507ca33383d37797b280ce259872f78a90e5  
969d81ffa92941064736f6c63430008000033



# 5. FINALLY DEPLOY IT TO DISPLAY THE OUTPUT

The screenshot displays the Remix Ethereum IDE interface within a web browser. The browser's address bar shows the URL: `remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js`. The interface is divided into several panels:

- Left Panel (Deploy & Run Transactions):** Contains a "CONTRACT" section with a dropdown menu showing "udhaya - contracts/mpm[sol]". Below it is a "Deploy" button and a checkbox for "Publish to IPFS". A "Load contract from Address" button is also present. The "Transactions recorded" section shows 6 transactions. The "Deployed Contracts" section lists a contract named "UDHAYA AT 0xDA0...42B53 (ME)". Below this, the "Balance" is shown as "0 ETH". A "zoon8" button is visible, with the input "0: string: zoon 4". The "Low level interactions" section shows the "CALLDATA" field and a "Transact" button.
- Center Panel (Code Editor):** Displays the Solidity code for the "mpm[sol]" contract. The code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract udhaya{
5     function zoon8() public pure returns(string memory){
6         return "zoon 4";
7     }
8 }
```
- Right Panel (Transaction Details):** Shows the details of a transaction. The "CALL" section indicates the transaction was from address `0x5B38Da6a701c568545dCfcB03Fc8875f56beddC4` to the contract `udhaya.zoon8()` with data `0xcc5...ba502`. The "from" field shows the sender's address, and the "to" field shows the contract address. The "execution cost" is listed as "741 gas (Cost only applies when called by a contract)". The "input" field shows the data `0xcc5...ba502`. The "decoded input" is an empty object `{}`. The "decoded output" is an array containing the string `"0": "string: zoon 4"`. The "logs" section is empty.

The Windows taskbar at the bottom shows the system clock as 8:58 PM on 10/13/2023, with the language set to ENG IN.