

# ASSIGNMENT 1

NAME	P.SANDHYA
ZONE	4
COLLEGE	GKM COLLEGE OF ENGINEERING AND TECHNOLOGY

# 1. GO TO THE CHROME ATFORM OPEN REMIX PLARTF

The screenshot shows the Remix Ethereum IDE interface within a Chrome browser window. The browser's address bar displays the URL `remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js`. The IDE interface is dark-themed and includes a sidebar on the left with icons for Home, Files, Learn, and a search bar. The main workspace is divided into several sections: a 'Featured' section with a 'BETA TESTING' announcement and a 'Sign up' button; a 'Get Started - Project Templates' section with four options: 'GNOSIS SAFE MULTISIG', 'OXPROJECT ERC20', 'OPENZEPPELIN ERC20', and 'OPENZEPPELIN ERC721'; and a 'Featured Plugins' section with four options: 'SOLIDITY ANALYZERS', 'COOKBOOK', 'SOLIDITY', and 'SOURCIFY'. The bottom of the interface features a status bar with a search bar, a 'listen on all transactions' checkbox, and a system tray showing the time (17:20) and date (15-10-2023).

Remix - Ethereum IDE x WhatsApp x +

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js

Gmail YouTube Maps

REMIX

The Native IDE for Web3 Development.

Website Documentation Remix Plugin Remix Desktop

Search Documentation

Files

New File Open File Access File System

Load from

GitHub Gist IPFS HTTPS

Learn

Remix Basics

An introduction to Remix's interface and basic operations.

Get Started

Intro to Solidity

Featured

BETA TESTING

Our community supports us.

Help us beta test releases now and get a handle on new features!

Sign up

Get Started - Project Templates

GNOSIS SAFE MULTISIG

Create Multi-Signature wallets using this template.

OXPROJECT ERC20

Create an ERC20 token by importing OxProject contract.

OPENZEPPELIN ERC20

Create an ERC20 token by importing OpenZeppelin library.

OPENZEPPELIN ERC721

Create an NFT token by importing OpenZeppelin

Featured Plugins

SOLIDITY ANALYZERS

Analyze your code using Remix, Solhint and Slither.

COOKBOOK

Find smart contracts, solidity libraries, and discover

SOLIDITY

Compile, test, and analyze smart contracts.

SOURCIFY

Solidity contract and met verification service.

listen on all transactions

Search with transaction hash or address

Type here to search

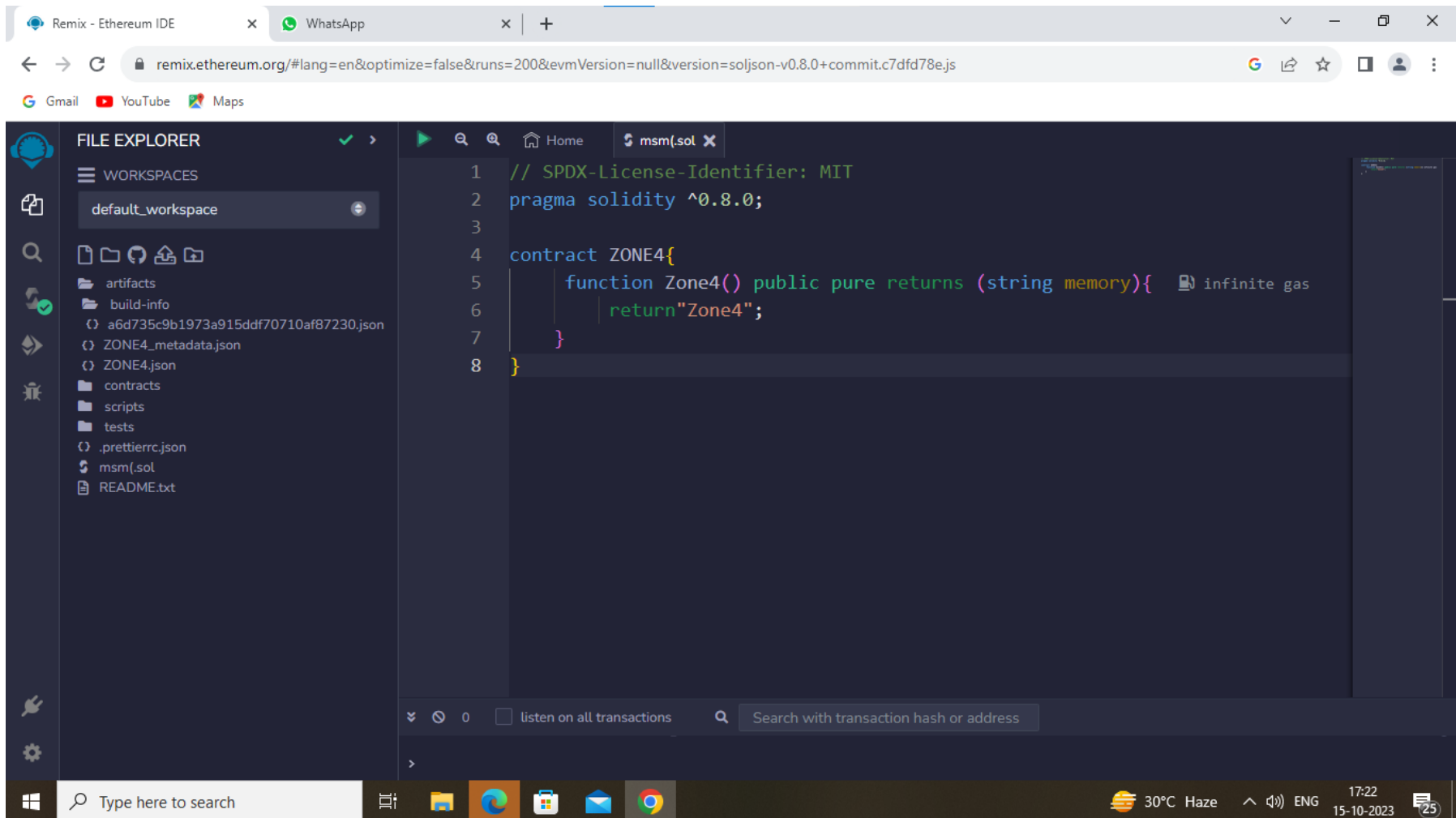
30°C Haze 17:20 15-10-2023

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

The screenshot shows the Remix Ethereum IDE interface in a web browser. The browser's address bar displays 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:

- FILE EXPLORER:** Located on the left, it shows a file tree for the 'default\_workspace'. The files listed are: `artifacts`, `build-info`, `a6d735c9b1973a915ddf70710af87230.json`, `ZONE4_metadata.json`, `ZONE4.json`, `contracts`, `scripts`, `tests`, `.prettierrc.json`, `msm.sol` (selected), and `README.txt`.
- REMIX Header:** The top center of the IDE features the 'REMIX' logo, social media links, and the tagline 'The Native IDE for Web3 Development.' Below this is a 'Search Documentation' bar.
- Files Panel:** Below the header, there are buttons for 'New File' and 'Open File', and a section for 'Access File System' with options to 'Load from' GitHub, Gist, IPFS, or HTTPS.
- Learn Section:** A 'Learn' section with a 'Remix Basics' link, described as 'An introduction to Remix's interface and basic operations.'
- Featured Section:** A 'Featured' section with a 'WATCH TO LEARN' video tip from the Remix Team. It includes a 'Watch' button and text stating: 'Remix has a growing library of videos containing lots of tips for using the tool. Check them out and subscribe to get our latest uploads.'
- Get Started - Project Templates:** A section with three templates: 'GNOSIS SAFE MULTISIG' (Create Multi-Signature wallets using this template), 'OXPROJECT ERC20' (Create an ERC20 token by importing OxProject contract), and 'OPENZEPELIN ERC20' (Create an ERC20 token by importing OpenZeppelin lib).
- Featured Plugins:** A section with three plugins: 'SOLIDITY ANALYZERS' (Analyze your code using Remix, Solhint and Slither), 'COOKBOOK' (Find smart contracts, solidity libraries, and discover), and 'SOLIDITY' (Compile, test, and analyze smart contracts).
- Bottom Panel:** At the bottom, there is a search bar with the text 'Search with transaction hash or address', a checkbox for 'listen on all transactions', and a status bar showing '30°C Haze', 'ENG', and the date '15-10-2023'.

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



The screenshot displays the Remix Ethereum IDE interface. The top browser window shows the URL `remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js`. The left sidebar contains a 'FILE EXPLORER' with a 'WORKSPACES' section showing 'default\_workspace' and a file list including 'artifacts', 'build-info', 'a6d735c9b1973a915ddf70710af87230.json', 'ZONE4\_metadata.json', 'ZONE4.json', 'contracts', 'scripts', 'tests', '.prettierrc.json', 'msm(.sol)', and 'README.txt'. The main editor area shows a Solidity program in a file named 'msm.sol'. The code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract ZONE4{
5     function Zone4() public pure returns (string memory){ infinite gas
6         return "Zone4";
7     }
8 }
```

The bottom status bar shows '0' transactions, a search bar with the text 'Search with transaction hash or address', and system information including '30°C Haze', 'ENG', and the date '15-10-2023'.

FILE EXPLORER

WORKSPACES

default\_workspace

artifacts

build-info

a6d735c9b1973a915ddf70710af87230.json

ZONE4\_metadata.json

ZONE4.json

contracts

scripts

tests

.prettierrc.json

msm(.sol

README.txt

1 // SPDX-License-Identifier: MIT

2 pragma solidity ^0.8.0;

3

4 contract ZONE4{

5 | function Zone4() public pure returns (string memory){ infinite gas

6 | | return"Zone4";

7 | }

8 }

listen on all transactions

Search with transaction hash or address

CALL [call] from: 0x5B38Da6a701c568545dCfc8B03Fc8875f56beddC4 to: ZONE4.Zone4() data: 0xc9b...e1928

from 0x5B38Da6a701c568545dCfc8B03Fc8875f56beddC4

to ZONE4.Zone4() 0xd9145CCE52D386f254917e481eB44e9943F39138

execution cost 741 gas (Cost only applies when called by a contract)

input 0xc9b...e1928

decoded input {}

decoded output { "0": "string: Zone4" }

Type here to search

📁 🌐 📁 📧 🌐

🌤 30°C Haze 📶 🔊 ENG 17:25 15-10-2023 25

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

ABI

```
[
  {
    "inputs": [],
    "name": "Zone4",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "pure",
    "type": "function"
  }
]
```

# BYTECODE

608060405234801561001057600080fd5b5061017c806100206000396  
000f3fe608060405234801561001057600080fd5b506004361061002b5  
760003560e01c8063c9be192814610030575b600080fd5b6100386100  
4e565b60405161004591906100c4565b60405180910390f35b6060604  
0518060400160405280600581526020017f5a6f6e6534000000000000  
00081525090509056  
5b6000610096826100e6565b6100a081856100f1565b93506100b0818  
560208601610102565b6100b981610135565b8401915050929150505  
65b600060208201905081810360008301526100de818461008b565b9  
05092915050565b600081519050919050565b6000828252602082019  
05092915050565b60005b83811015610120578082015181840152602  
081019050610105565b8381111561012f576000848401525b50505050  
565b6000601f19601f830116905091905056fea2646970667358221220  
9f1f861fcf66d913a9cdbb6ea4738784cf8723ef59258eb10998f376ae76  
458d64736f6c63430008000033

# 5. FINALLY DEPLOY IT TO DISPLAY THE OUTPUT

The screenshot displays the Remix Ethereum IDE interface. The top bar shows the browser address: `remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js`. The left sidebar contains the FILE EXPLORER with a tree view of the workspace, including folders like `artifacts`, `build-info`, `contracts`, and `scripts`, as well as files like `ZONE4_metadata.json`, `ZONE4.json`, `msm.sol`, and `README.txt`.

The main editor displays the Solidity code for `msm.sol`:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract ZONE4{
5     function Zone4() public pure returns (string memory){ infinite gas
6         return "Zone4";
7     }
8 }
```

Below the code editor, the DEBUG console shows the execution details of a `CALL` transaction:

- CALL** [call] from: `0x5B38Da6a701c568545dCfcB03FcB875f56beddC4` to: `ZONE4.Zone4()` data: `0xc9b...e1928`
- from**: `0x5B38Da6a701c568545dCfcB03FcB875f56beddC4`
- to**: `ZONE4.Zone4() 0xd9145CCE52D386f254917e481e844e9943F39138`
- execution cost**: 741 gas (Cost only applies when called by a contract)
- input**: `0xc9b...e1928`
- decoded input**: `{}`
- decoded output**: `{ "0": "string: Zone4" }`

The bottom status bar shows the system clock at 17:29 on 15-10-2023, along with icons for Live, ENG, and a notification badge.



 Search with transaction hash or address

```
decoded output      {
                    "0": "string: Zone4"
                    } [D]
```