

## Assignment 1

NAME PRASANNA.M	
ZONE	2
COLLEGE	APOLLO ENGINEERING COLLEGE

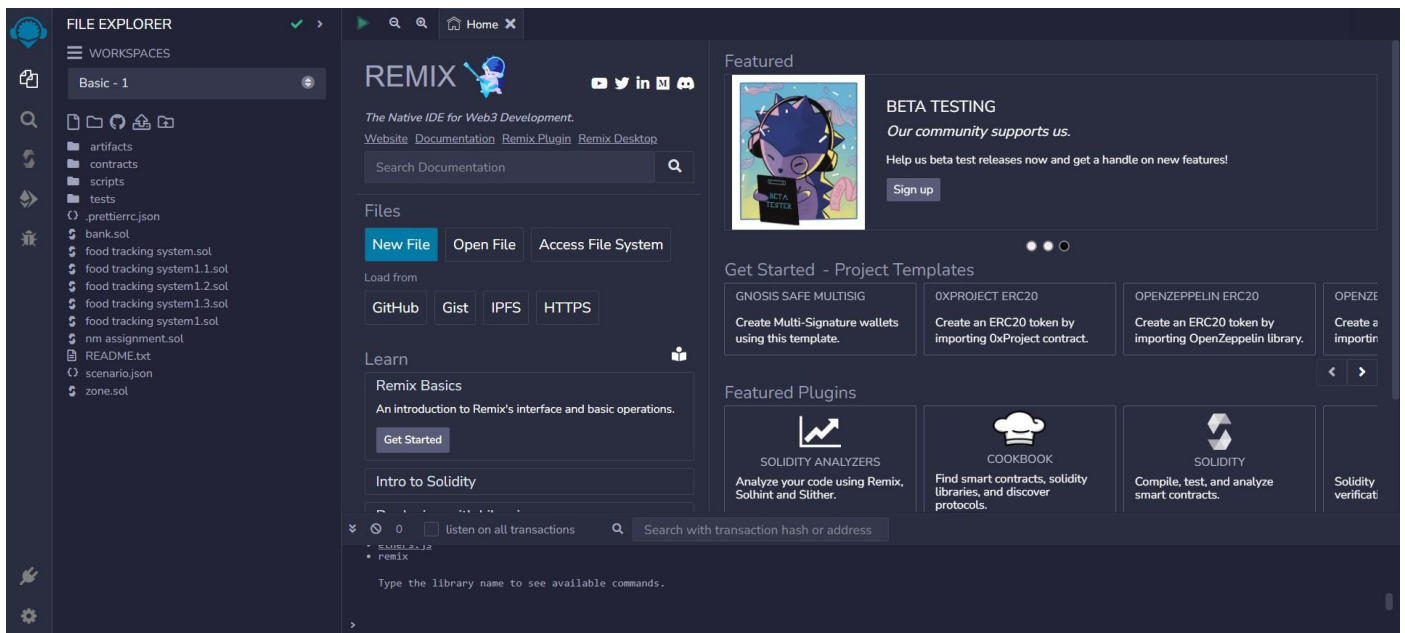
# 1. GO TO THE CHROME ATFORM OPEN REMIXPLARTFORM

The screenshot displays the Remix IDE interface. At the top, there's a navigation bar with tabs for 'Home', 'assignment.sol', and 'assignment1.sol'. The main area is divided into several sections:

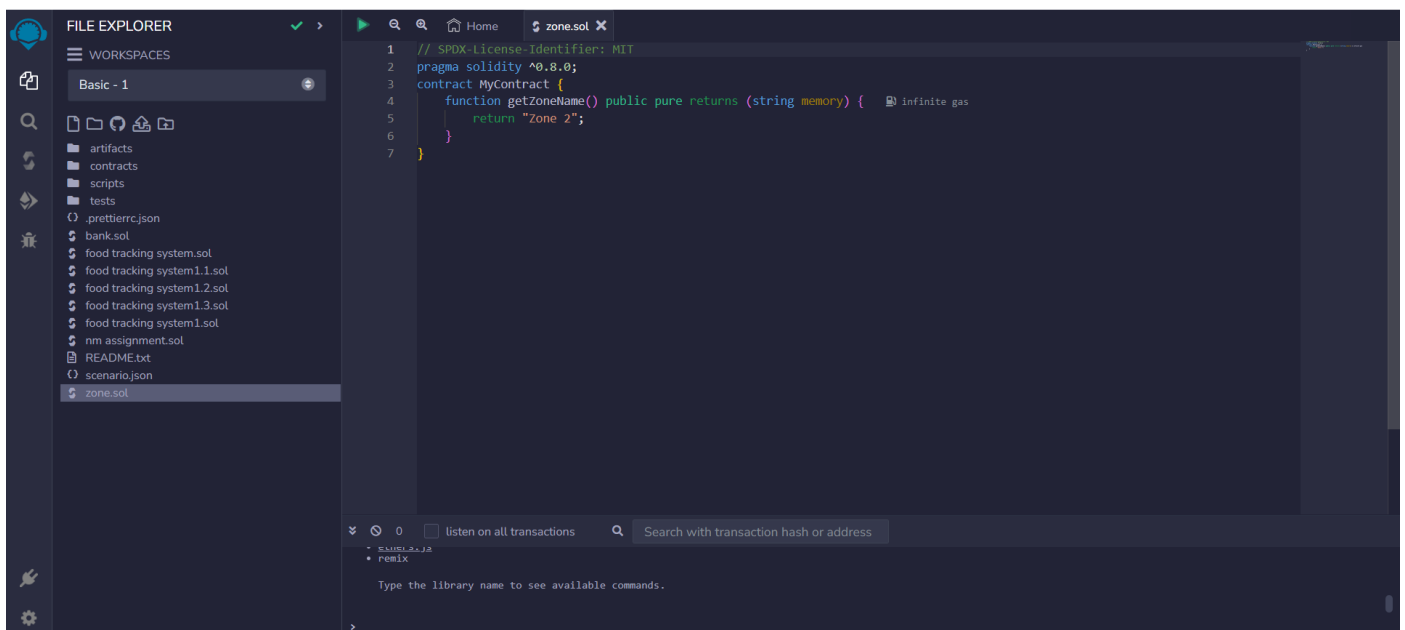
- Left Sidebar:** Contains the 'REMIX' logo, social media links, a search bar for documentation, and a 'Files' section with buttons for 'New File', 'Open File', and 'Access File System'. Below this is a 'Learn' section with links to 'Remix Basics', 'Intro to Solidity', and 'Deploying with Libraries'.
- Featured Section:** Promotes 'BETA TESTING' with a 'Sign up' button.
- Get Started - Project Templates:** Offers several templates for creating tokens and wallets, including 'GNOSIS SAFE MULTISIG', '0XPROJECT ERC20', 'OPENZEPPELIN ERC20', 'OPENZEPPELIN ERC721', and 'OPENZ'.
- Featured Plugins:** Lists plugins like 'SOLIDITY ANALYZERS', 'COOKBOOK', 'SOLIDITY', and 'SOURCIFY', each with a brief description of their functionality.
- Scam Alert:** A warning section stating that the only URL Remix uses is 'remix.ethereum.org' and advising users to be wary of online videos promoting 'liquidity front runner bots'.

At the bottom, there's a status bar with a search bar and a checkbox for 'listen on all transactions'.

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




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



The screenshot shows the Visual Studio Code editor interface. On the left, the 'FILE EXPLORER' sidebar displays a workspace named 'Basic - 1' with a file tree containing various files and folders. The file 'zone.sol' is selected. The main editor area shows the content of 'zone.sol', which is a Solidity contract named 'MyContract'. The contract includes a pragma statement for Solidity version 0.8.0 and a function 'getZoneName()' that returns the string 'Zone 2'. The bottom status bar shows the Remix logo and a search bar for transaction hashes or addresses.

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



SOLIDITY COMPILER

COMPILER +

0.8.18+commit.87f61d96

Include nightly builds

Auto compile

Hide warnings

Advanced Configurations

Compile zone.sol

Compile and Run script

CONTRACT

MyContract (zone.sol)

Publish on Ipfs

Publish on Swarm

Compilation Details

ABI

Bytecode

zone.sol

1 // SPDX-License-Identifier: MIT

2 pragma solidity ^0.8.0;

3 contract MyContract {

4     function getZoneName() public pure returns (string memory) { infinite gas

5         return "Zone 2";

6     }

7 }


0

listen on all transactions

Search with transaction hash or address

remix

Type the library name to see available commands.



SOLIDITY COMPILER

COMPILER +

0.8.18+commit.87f61d96

Include nightly builds

Auto compile

Hide warnings

Advanced Configurations

Compile zone.sol

Compile and Run script

CONTRACT

MyContract (zone.sol)

Publish on Ipfs

Publish on Swarm

Compilation Details

ABI

Bytecode

zone.sol

1 // SPDX-License-Identifier: MIT

2 pragma solidity ^0.8.0;

3 contract MyContract {

4     function getZoneName() public pure returns (string memory) { infinite gas

5         return "Zone 2";

6     }

7 }

0

listen on all transactions

Search with transaction hash or address

remix

Type the library name to see available commands.

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

## ABI:

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

## BYTECODE:

[illegible]

# FINALLY DEPLOY IT TO DISPLAY THE OUTPUT

The screenshot displays the Remix IDE interface, which is used for developing and deploying smart contracts. The interface is divided into several panels:

- DEPLOY & RUN TRANSACTIONS:** This panel on the left contains settings for deploying a contract. It includes fields for the account (0x5B3...eddC4), gas limit (3000000), and value (0 Wei). The contract selected is "MyContract - zone.sol". There are buttons for "Deploy", "Publish to IPFS", and "At Address".
- Transactions recorded:** This section shows a list of transactions, including "MYCONTRACT AT 0XD91...39138" and "MYCONTRACT AT 0XD8B...33FA8".
- Code Editor:** The central area shows the Solidity code for "zone.sol". The code defines a contract "MyContract" with a function "getZoneName" that returns the string "Zone 2".
- Console:** The bottom panel shows the output of the deployment, including the message "Welcome to Remix 0.36.3" and "Your files are stored in indexedDB, 10.06 KB / 168.46 GB used".

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