

**E.INDHUMATHI**

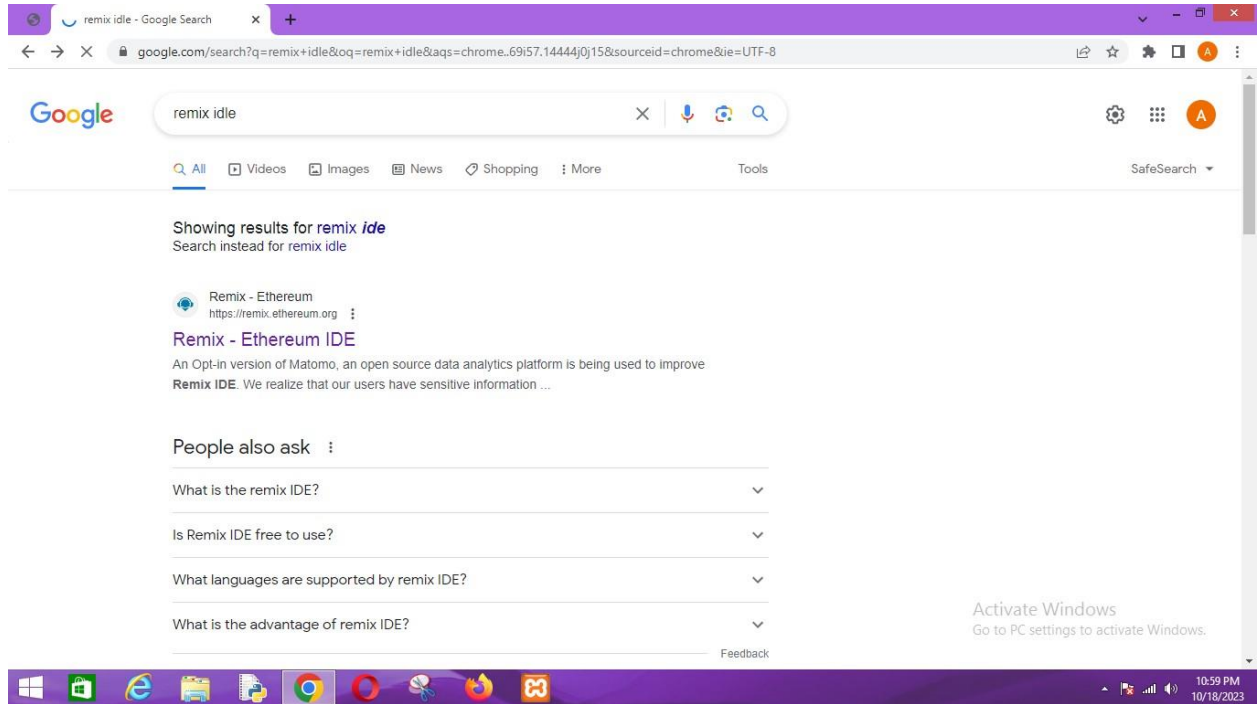
**NM ID: IFE25C7903C885DB3F441C0E28823E3B**

**BLOCK CHAIN**

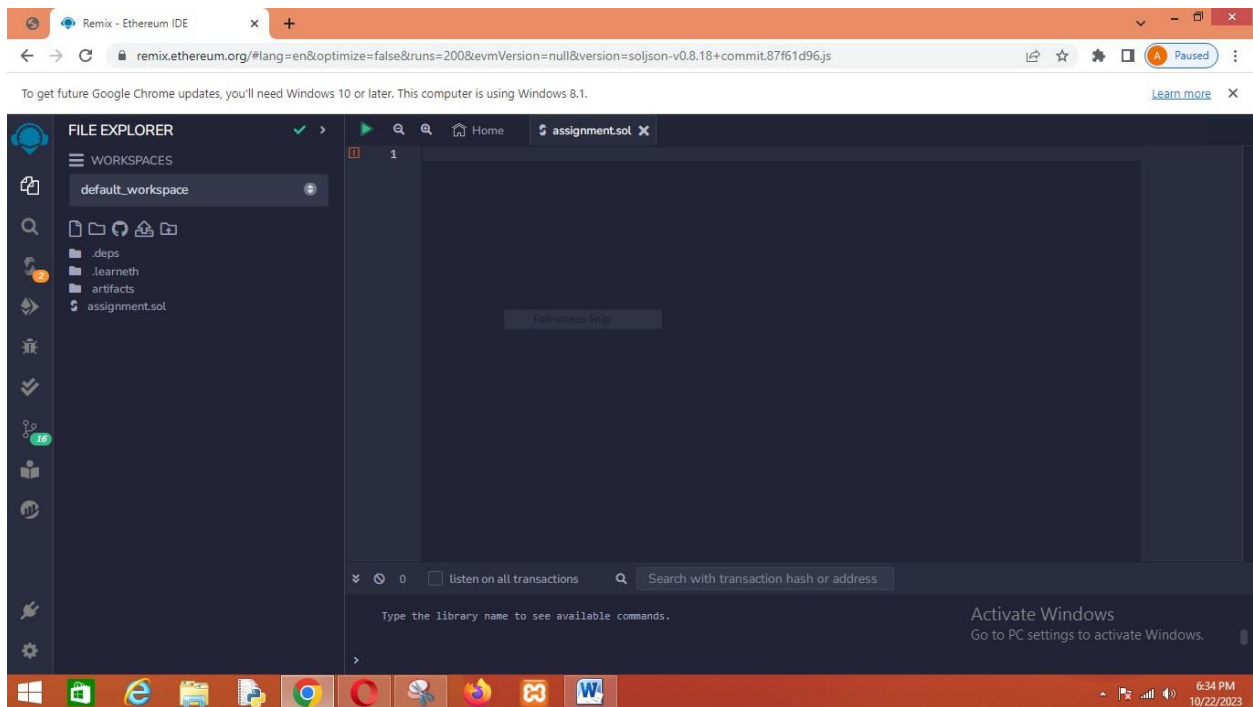
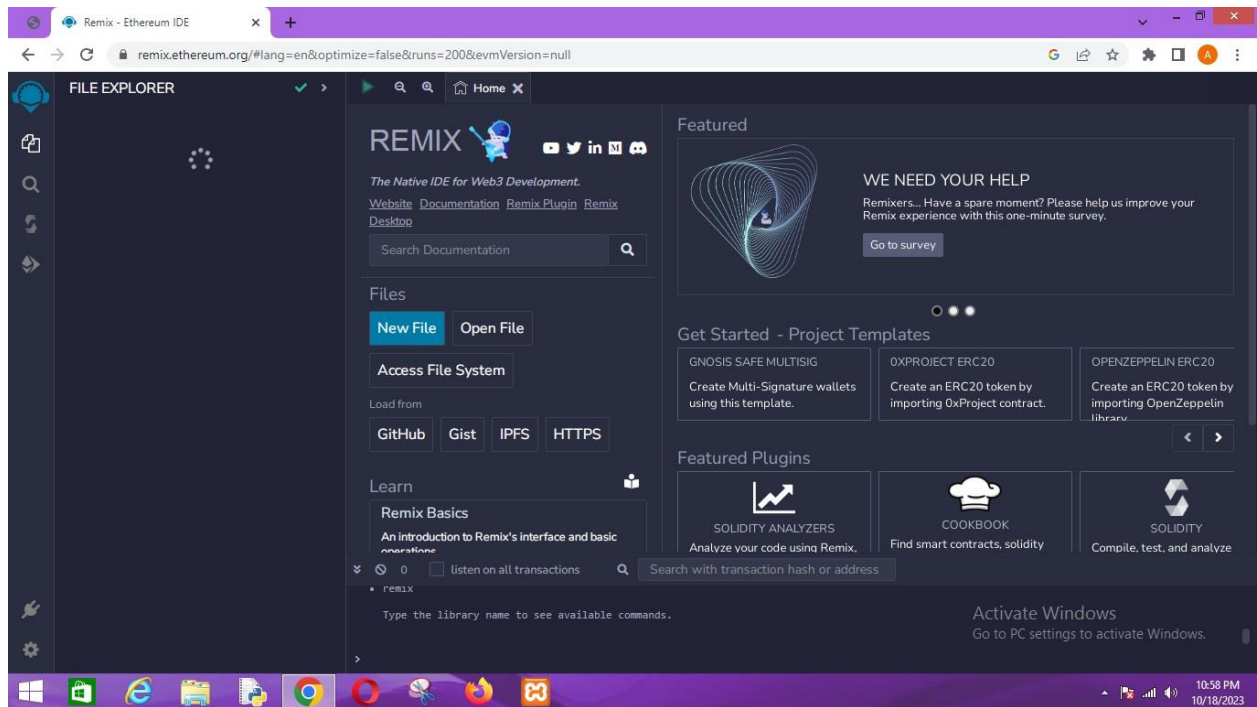
**ASSIGNMENT**

**SRI RAAJA RAAJAN COLLEGE OF ENGINEERING AND TECHNOLOGY**

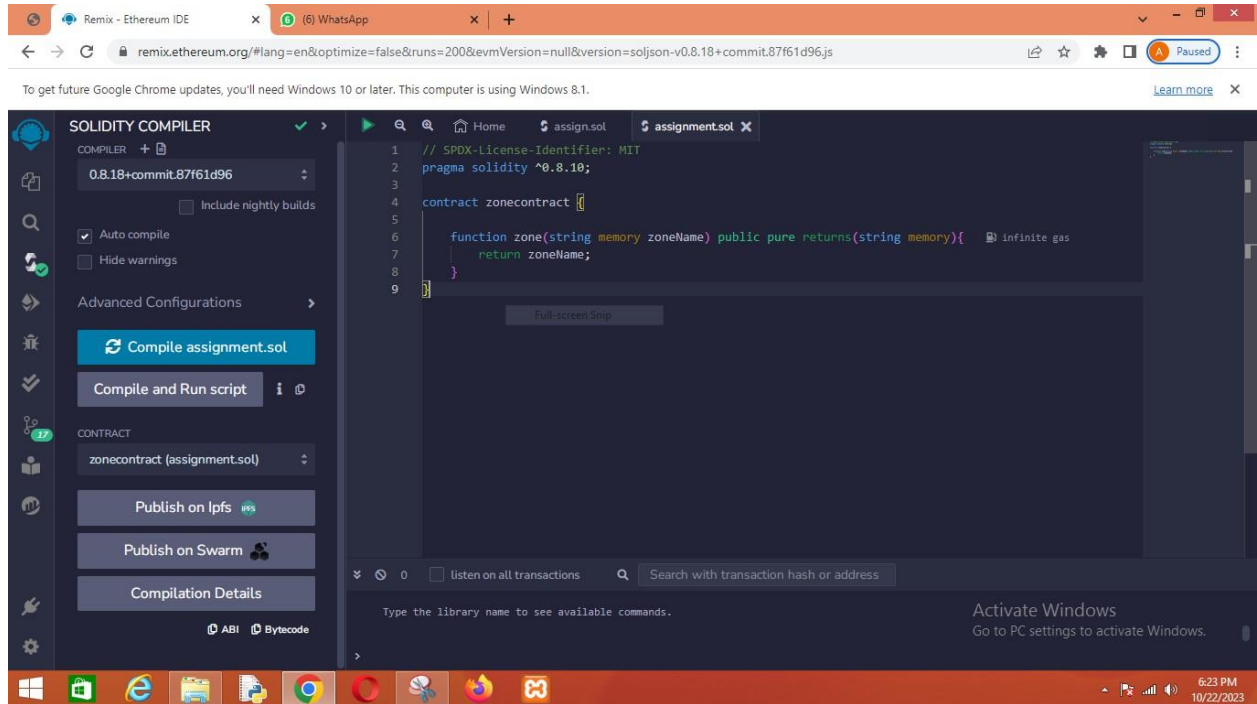
# 1.Go to the chrome and open remix platform



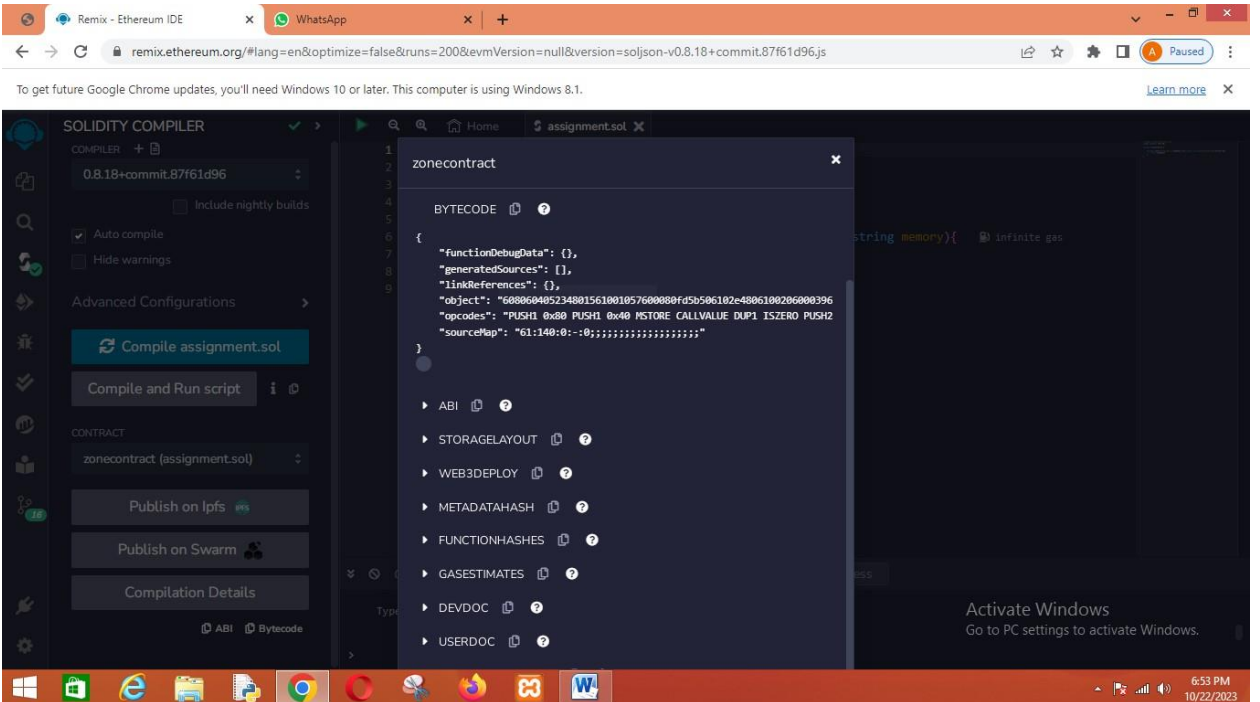
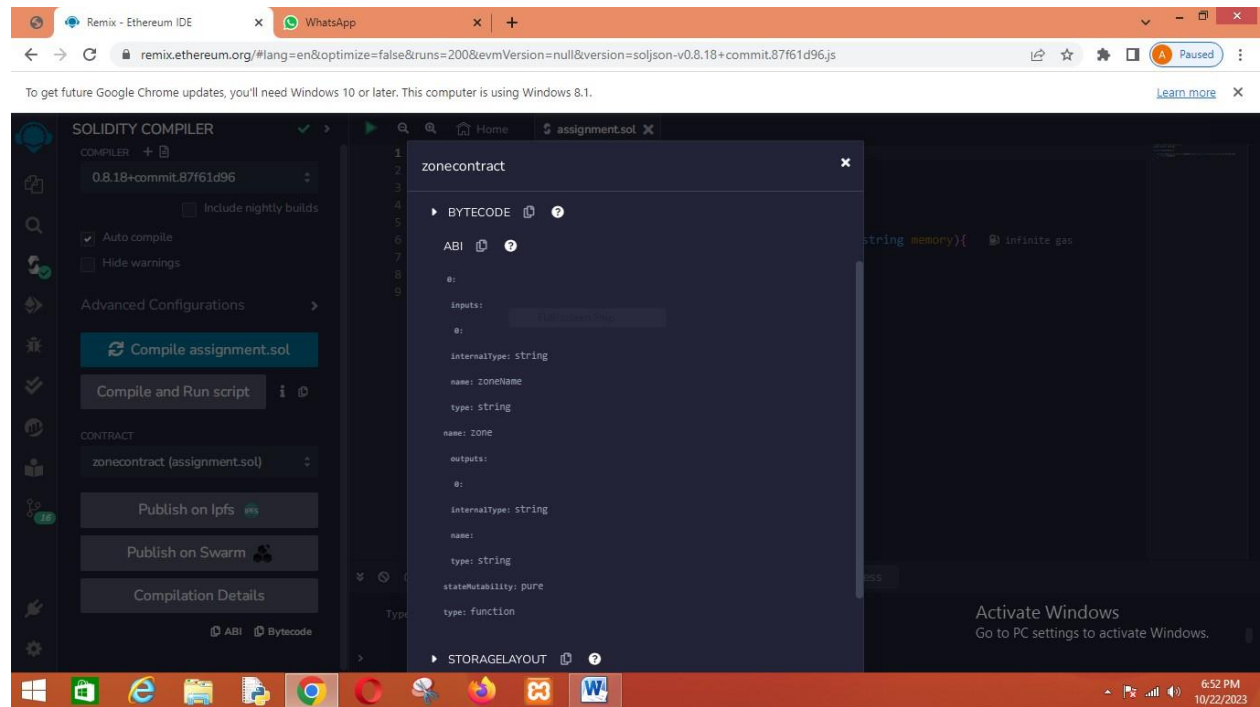
## 2. Open the remix page and create a new file.



### 3. Create a program to return your string “zone name”.



## 4. Compile it to get API and BYTECODE



## 5. Deploy it to display the output :

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

- Left Panel (Environment):** Shows the current environment (Remix VM (Shanghai)), account (0x5B3...eddC4), gas limit (3000000), and value (0 Wei). It also displays the contract name (zonecontract - assignment.sol) and the EVM version (paris).
- Top Panel (Code Editor):** Contains the Solidity code for the `zonecontract`. The code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.10;
3
4 contract zonecontract {
5
6     function zone(string memory zoneName) public pure returns(string memory){ infinite gas
7         return zoneName;
8     }
9 }
```
- Bottom Panel (Debugger):** Shows the execution state of the contract. It includes a "Function Stack" and "Solidity Locals" section, both of which are currently empty. The "Step details" section shows the current step: "vm trace step: 0", "execution step: 0", "add memory", "gas: 3", "remaining gas: 180031", and "loaded address: (Contract Creation - Step 0)".

The bottom status bar indicates the current transaction hash and the gas used. The interface also includes a "Deploy" button and a "Debug" button. The bottom right corner shows the system clock (6:24 PM, 10/22/2023) and a Windows activation watermark.

**Thank you...**