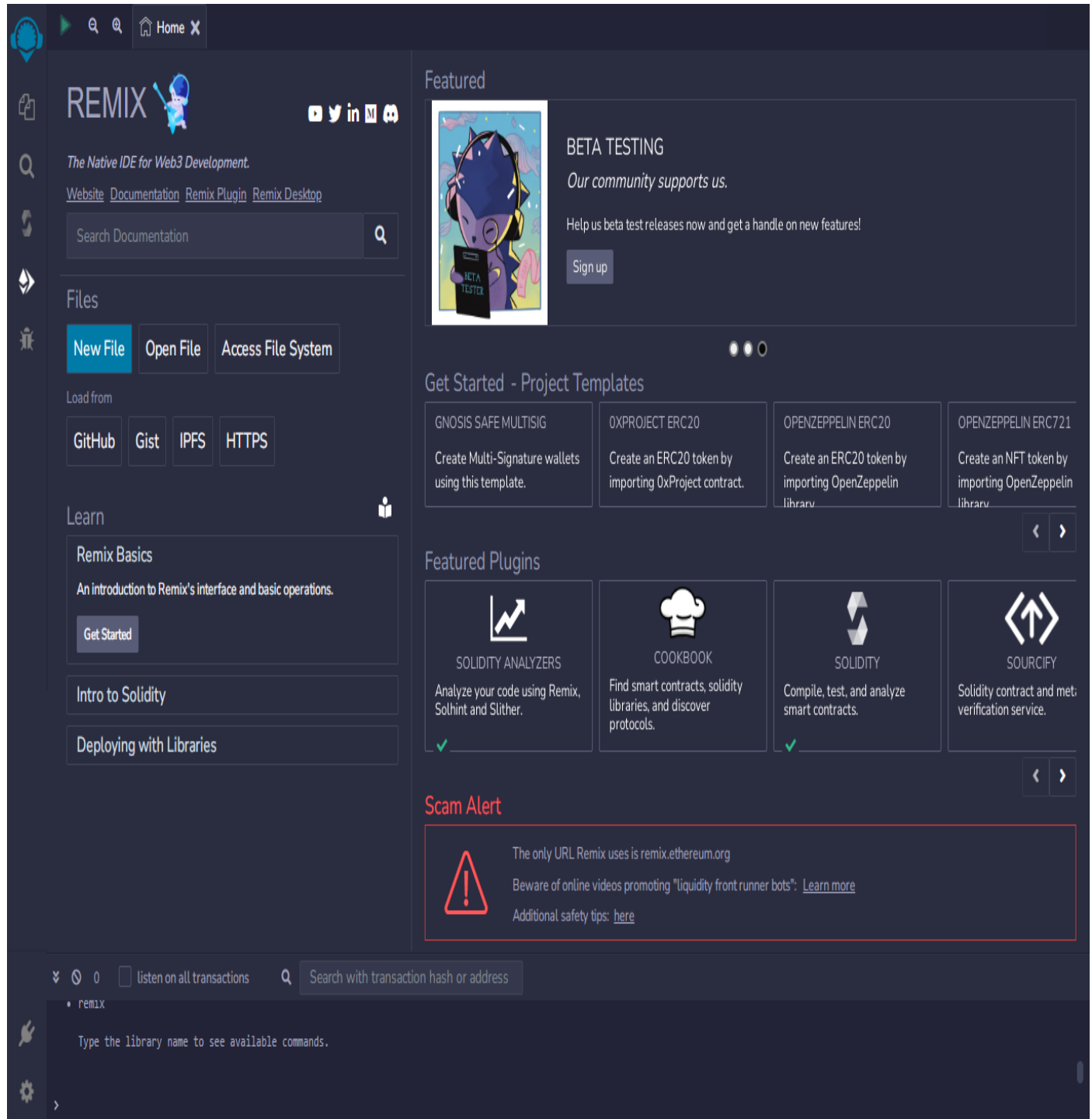


<b>NAME</b>	K.KAVIYA
<b>ZONE</b>	4
<b>COLLEGE</b>	GKM COLLEGE OF ENGINEERING AND TECHNOLOGY

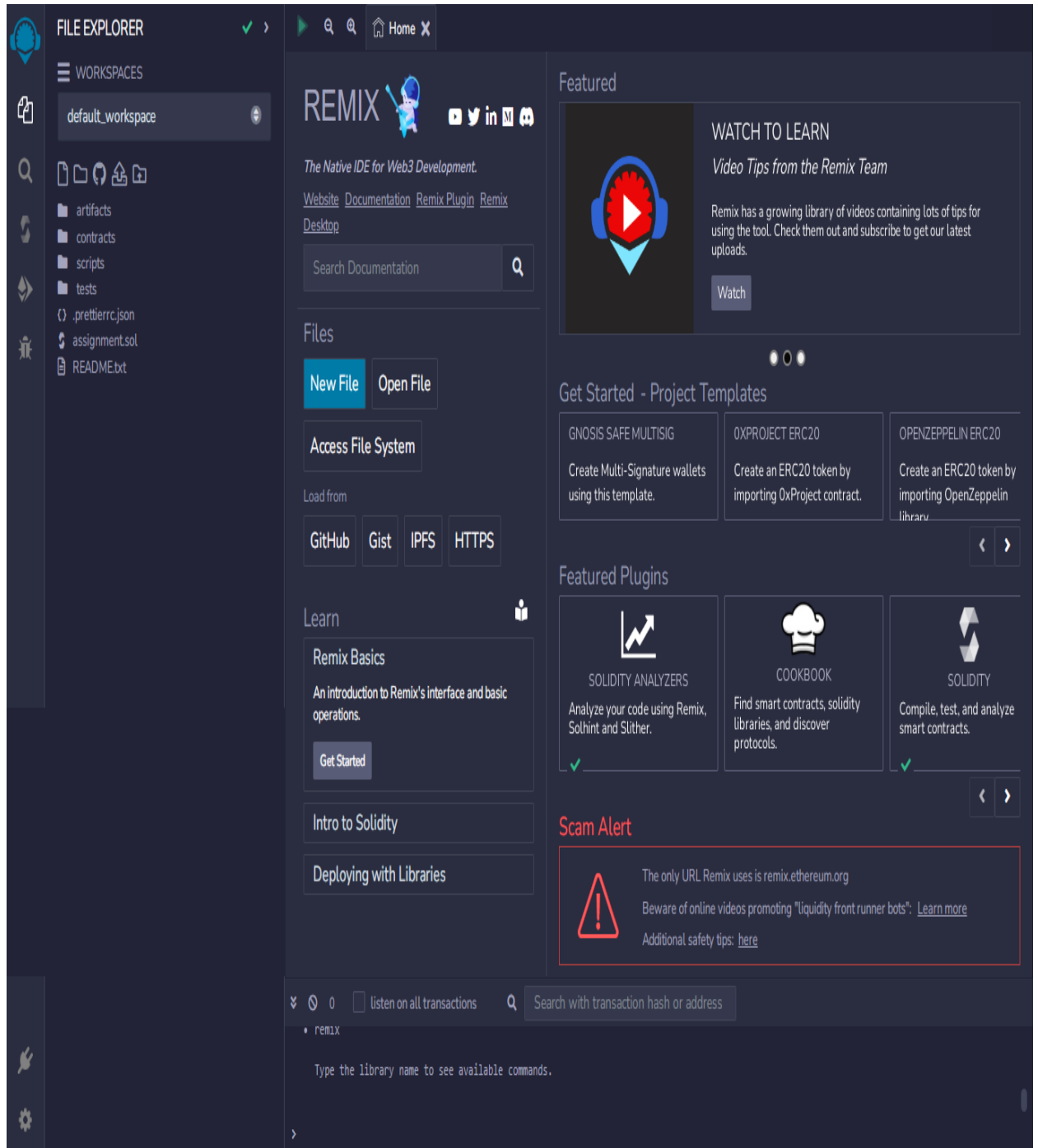
# 1. GO TO THE CHROME ATFORM OPEN REMIX PLARTFORM



The screenshot displays the Remix IDE interface, which is a web-based development environment for Ethereum. The interface is divided into several sections:

- Header:** Includes a search bar, a "Home" button, and social media links (YouTube, Twitter, LinkedIn, GitHub).
- Left Sidebar:**
  - REMIX:** The Native IDE for Web3 Development. Includes links to Website, Documentation, Remix Plugin, and Remix Desktop.
  - Search:** A search bar for documentation.
  - Files:** Buttons for "New File", "Open File", and "Access File System".
  - Load from:** Buttons for "GitHub", "Gist", "IPFS", and "HTTPS".
  - Learn:** A section with links to "Remix Basics", "Intro to Solidity", and "Deploying with Libraries".
- Main Content Area:**
  - Featured:** A section with a "BETA TESTING" announcement and a "Sign up" button.
  - Get Started - Project Templates:** A section with four templates: "GNOSIS SAFE MULTISIG", "OXPROJECT ERC20", "OPENZEPPELIN ERC20", and "OPENZEPPELIN ERC721".
  - Featured Plugins:** A section with four plugins: "SOLIDITY ANALYZERS", "COOKBOOK", "SOLIDITY", and "SOURCIFY".
  - Scam Alert:** A red warning box with a triangle icon, stating: "The only URL Remix uses is remix.ethereum.org. Beware of online videos promoting 'liquidity front runner bots': [Learn more](#). Additional safety tips: [here](#)".
- Bottom Bar:** Includes a "listen on all transactions" checkbox, a search bar for transaction hash or address, and a "remix" button.

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



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

The screenshot displays the Remix IDE interface. On the left sidebar, the 'SOLIDITY COMPILER' section is active, showing version 0.8.18+commit.87f61d96. Below the compiler settings, there are buttons for 'Compile assignment.sol', 'Compile and Run script', 'Publish on Ipfs', 'Publish on Swarm', and 'Compilation Details'. The main editor area shows a Solidity contract named 'ZoneNameContract' with a function 'getZoneName()' that returns the string 'Zone 4'. The bottom panel shows the terminal output, which includes a welcome message and a list of accessible libraries (web3 version 1.5.2, ethers.js, remix). The terminal also shows the execution of the 'getZoneName()' function, returning the value 'Zone 4'.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract ZoneNameContract {
    function getZoneName() public pure returns (string memory) {
        return "Zone 4";
    }
}
```

Welcome to Remix 0.36.3

Your files are stored in indexedDB, 131.26 KB / 87.6 GB used

You can use this terminal to:

- Check transactions details and start debugging.
- Execute JavaScript scripts:
  - Input a script directly in the command line interface
  - Select a Javascript file in the file explorer and then run `remix.execute()` or `remix.executeCurrent()` in the command line interface
  - Right click on a Javascript file in the file explorer and then click 'Run'

The following libraries are accessible:

- [web3 version 1.5.2](#)
- [ethers.js](#)
- [remix](#)

Type the library name to see available commands.

creation of ZoneNameContract pending...

[vm] from: 0x583...eddC4 to: ZoneNameContract.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x63e...4ffb6 **Debug**

call to ZoneNameContract.getZoneName

[call] from: 0x58380a6a701c568545dCfc803Fc8875F56beddC4 to: ZoneNameContract.getZoneName() data: 0x235...ed70f **Debug**

#### 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]

## 5. FINALLY DEPLOY IT TO DISPLAY THE OUTPUT

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

- Left Panel (Deploy & Run Transactions):** This panel contains settings for the deployment environment. It shows the environment as "Remix VM (Shanghai)", the VM type, the account address "0x5B3...eddC4 (99.99999999%)", the gas limit set to "3000000", and the value set to "0 Wei". The contract to be deployed is "ZoneNameContract - assignment.sol". There is a "Deploy" button and a checkbox for "Publish to IPFS". Below these, there is a section for "Transactions recorded" and a "Deployed Contracts" section. The "Deployed Contracts" section shows a contract named "ZONENAMECONTRACT AT 0XD91..." with a balance of "0 ETH". It includes a "getZoneName" button and a "Low level interactions" section with a "Transact" button.
- Center Panel (Code Editor):** This panel shows the Solidity code for the "assignment.sol" file. The code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract ZoneNameContract {
5     function getZoneName() public pure returns (string memory) { infinite gas
6         return "Zone 4";
7     }
8 }
```
- Right Panel (Debug Console):** This panel shows the execution results of the contract. It displays a call from "0x5B380a6a701c568545dcfc803Fc8875f56beddC4" to "ZoneNameContract.getZoneName()" with data "0x235...ed70f". A "Debug" button is visible next to the call.