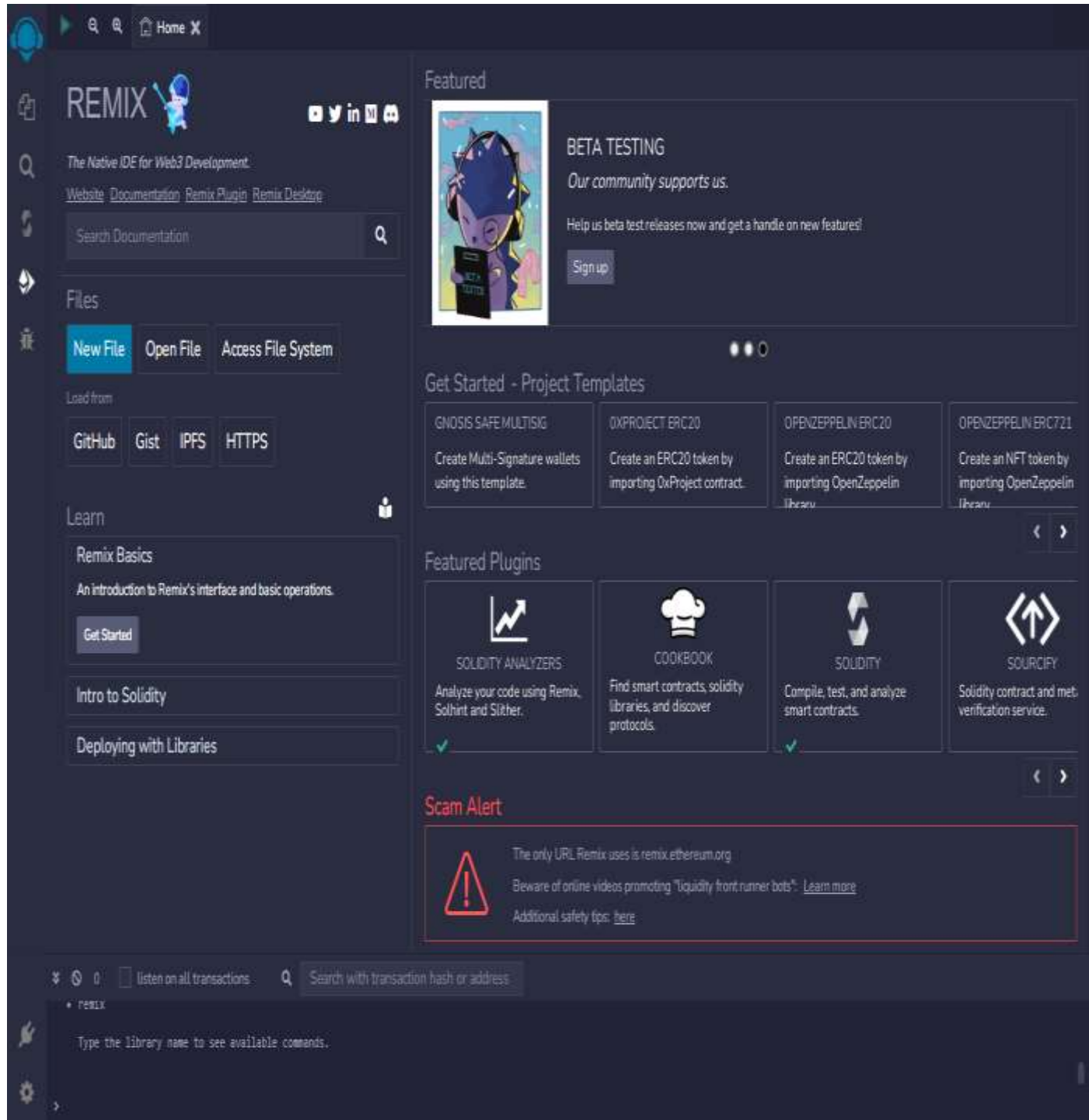
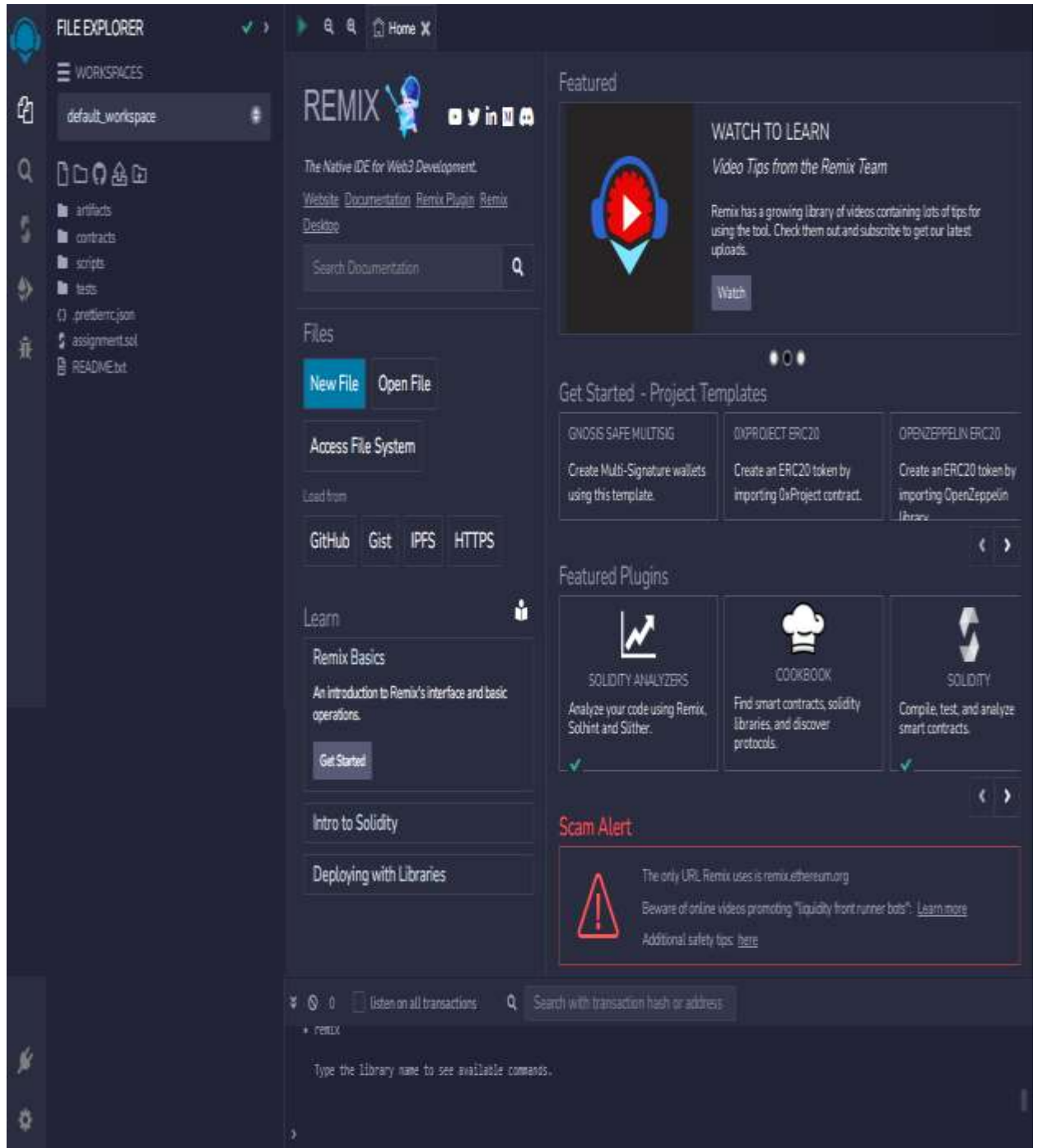


NAME	SENKATHIR.P
ZONE	4
COLLEGE	GKM COLLEGE OF ENGINEERING AND TECHNOLOGY

1. GO TO THE CHROME ATFORM OPEN REMIX PLARTFORM



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, the 'SOLIDITY COMPILER' sidebar shows the compiler version '0.8.18+commit.87f61d96' and options for 'Auto compile' and 'Hide warnings'. Below this, the 'CONTRACT' section lists 'ZoneNameContract (assignment.sol)'. The main editor area shows the following Solidity code:

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

The bottom panel shows the 'ABI' and 'Bytecode' tabs. The 'ABI' tab is active, displaying the contract's ABI. The 'Bytecode' tab is also visible. The bottom right corner shows the 'Debug' console with the following output:

```
[vm] from: 0x593...eddK4 to: ZoneNameContract.(constructor) value: 0 wei data: 0x686...28033 logs: 0 hash: 0x0fe...4f6f
call to ZoneNameContract.getZoneName
[call] from: 0x59330da6a781c568545dcf1863fc8675f568e0d4 to: ZoneNameContract.getZoneName() data: 0x235...ed70f
```

ABI:

5. FINALLY DEPLOY IT TO DISPLAY THE OUTPUT

The screenshot displays the Remix IDE interface, which is used for developing and deploying smart contracts. The main editor shows the Solidity code for the `ZoneNameContract`.

```
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 }
```

The left sidebar contains the "DEPLOY & RUN TRANSACTIONS" panel. It shows the environment set to "Remix VM (Shanghai)", the account address `0x583...addC4`, and the gas limit set to `3000000`. The contract selected is `ZoneNameContract - assignment.sol`. The "Deploy" button is highlighted.

The bottom panel shows the "Transactions recorded" section, indicating that a transaction has been recorded. The transaction details show a call to `ZoneNameContract.getZoneName()` from the account `0x5830a6a701c568545dCfc803fc875f568e0dC4` to the contract address `0x0001...`. The transaction data is `0x235...ed70f`.

The "Deployed Contracts" section shows the deployed contract `ZONENameCONTRACT AT 0x0001...`. The contract's balance is `0 ETH`. The `getZoneName` function is visible, and its output is `0: string Zone 4`.

The "Low level interactions" section shows the `CALLDATA` field, which is currently empty.