

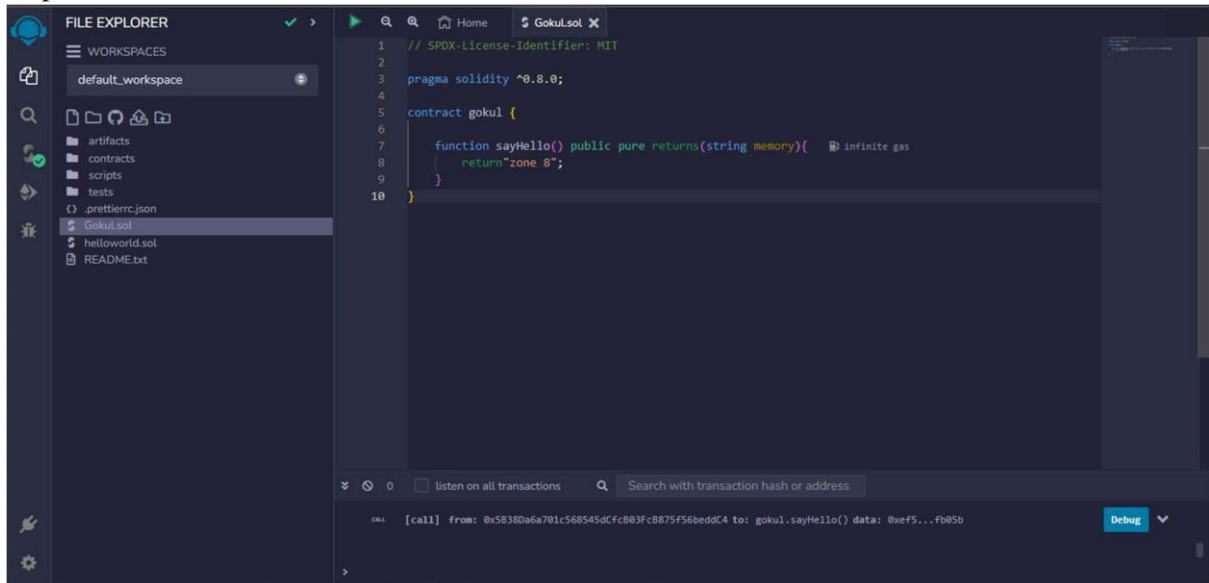
Assignment - I

BlockChain Technology

PROJECT NAME:	VACCINE TRACKING-TRANSPARENT
TEAM ID	NM2023TMID11923

Output :

Step 1 :



The screenshot shows a Solidity IDE interface. On the left, the 'FILE EXPLORER' panel displays a workspace named 'default_workspace' containing files: 'artifacts', 'contracts', 'scripts', 'tests', '.prettierrc.json', 'Goku.sol', 'helloworld.sol', and 'README.txt'. The main editor area shows the code for 'Goku.sol':

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 contract gokul {
6
7     function sayHello() public pure returns(string memory){ infinite gas
8         return "zone 8";
9     }
10 }
```

At the bottom, the 'DEBUG CONSOLE' panel shows a transaction call: '[call] from: 0x58380a6a701c568545dcfc803fc8875f56beddC4 to: gokul.sayHello() data: 0xef5...fb05b'. A 'Debug' button is visible on the right.

Step 2 :

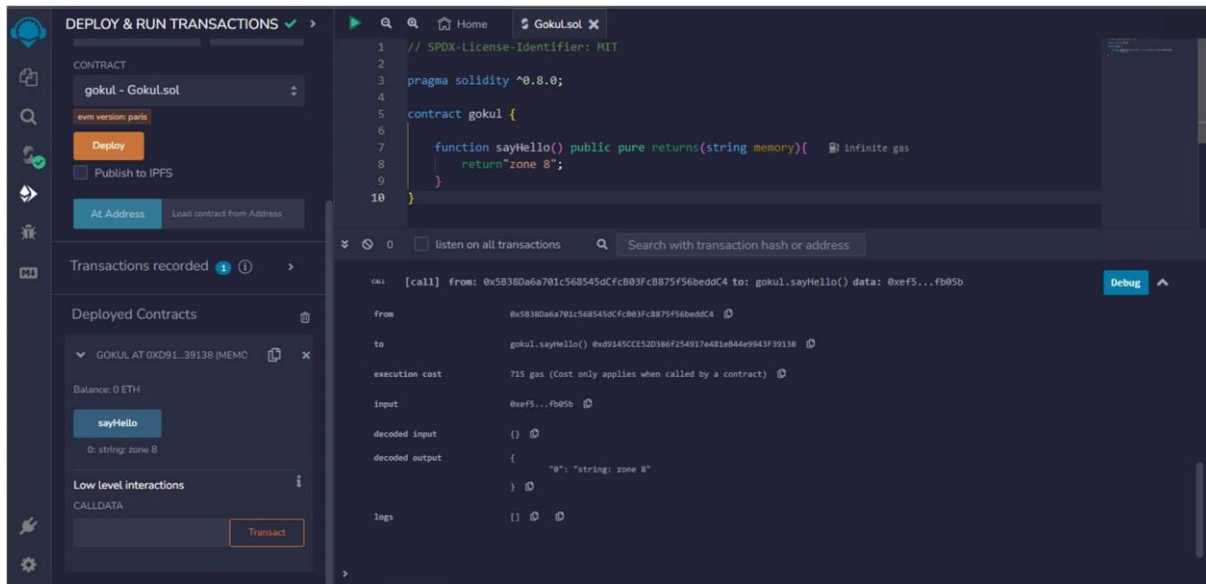
The screenshot shows the Solidity Compiler interface. On the left, the 'COMPILER' section is active, displaying version '0.8.18+commit.87f61d96'. Below this, there are checkboxes for 'Include nightly builds', 'Auto compile', and 'Hide warnings'. The 'Advanced Configurations' section is expanded, showing a 'Compile Gokul.sol' button and a 'Compile and Run script' button. The 'CONTRACT' section shows 'gokul (Gokul.sol)' selected, with buttons for 'Publish on Ipfs', 'Publish on Swarm', and 'Compilation Details'. The main editor area displays the Solidity code for 'Gokul.sol':

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 contract gokul {
6
7     function sayHello() public pure returns(string memory){ @infinite gas
8         return "zone 8";
9     }
10 }
```

At the bottom, the 'DEBUG' section shows a transaction log with a single entry: '[call] from: 0x58380a6a701c568545dcfc803fc8875f56beddC4 to: gokul.sayHello() data: 0xef5...fb05b'. A 'Debug' button is visible next to the entry.

Step 3 :

The screenshot shows the 'DEPLOY & RUN TRANSACTIONS' interface. On the left, the 'CONTRACT' section shows 'gokul - Gokul.sol' selected. Below this, there is a 'Deploy' button and a 'Publish to IPFS' checkbox. The 'At Address' section is expanded, showing a 'Load contract from Address' button. The 'Transactions recorded' section shows a list of transactions, including one for 'GOKUL AT 0XD91_3913B (MEMC)'. The 'Deployed Contracts' section shows a list of contracts, including 'GOKUL AT 0XD91_3913B (MEMC)'. The 'Low level interactions' section shows a 'Transact' button. The main editor area displays the same Solidity code as in the previous screenshot. The 'DEBUG' section shows a transaction log with a single entry: '[call] from: 0x58380a6a701c568545dcfc803fc8875f56beddC4 to: gokul.sayHello() data: 0xef5...fb05b'. A 'Debug' button is visible next to the entry.



ABI :

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

Bytecode :

608060405234801561001057600080fd5b50610173806100206000396000f3fe608060405234801561
001057600080fd5b506004361061002b5760003560e01c8063ef5fb05b14610030575b600080fd5b61
003861004e565b604051610045919061011b565b60405180910390f35b60606040518060400160405
280600681526020017f7a6f6e65203800
00815250905090565b600081519050919050565b600082825260208201905092915050565b60005b
838110156100c55780820151818401526020810190506100aa565b600084840152505050565b600
0601f19601f8301169050919050565b60006100ed8261008b565b6100f78185610096565b93506101
078185602086016100a7565b610110816100d1565b840191505092915050565b6000602082019050
818103600083015261013581846100e2565b90509291505056fea2646970667358221220bc9d72eb3
d398ab6f051889207e3d105a62e32be843e77f5b7c5383e894afd3164736f6c63430008120033