Name: - Parth Patel

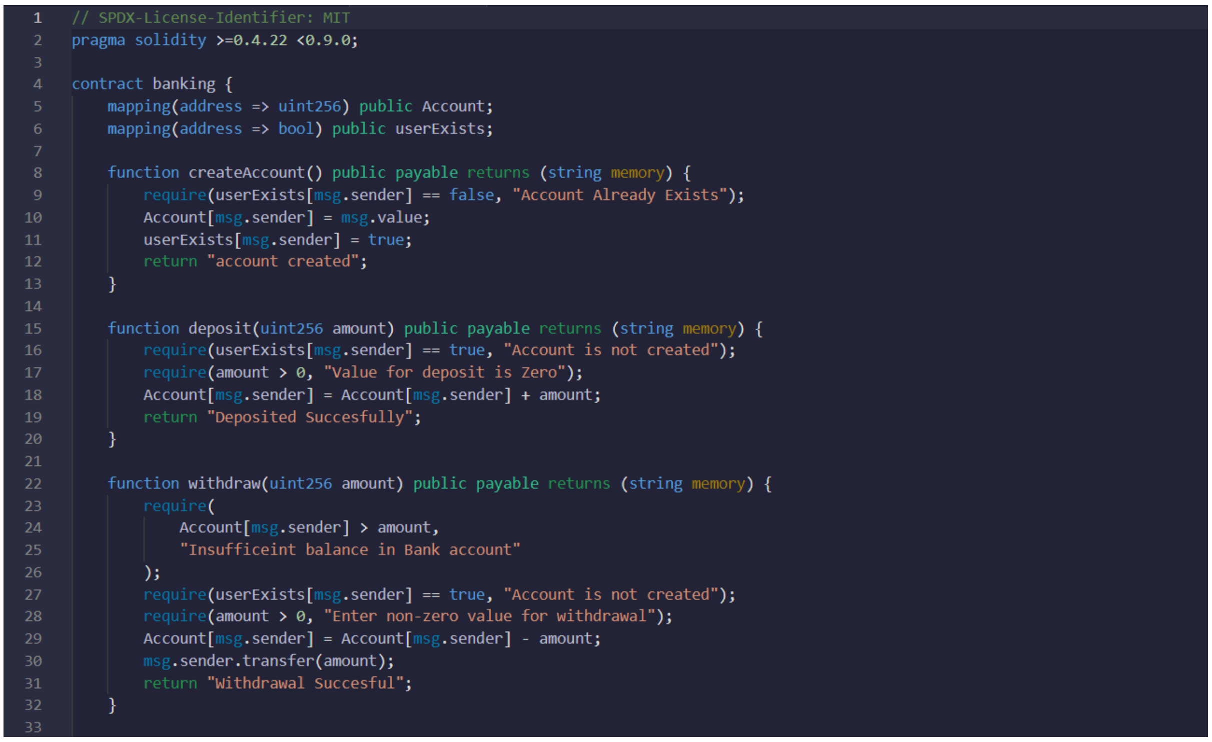
Roll Number: - 19BCP091

Subject: - Blockchain Technology lab

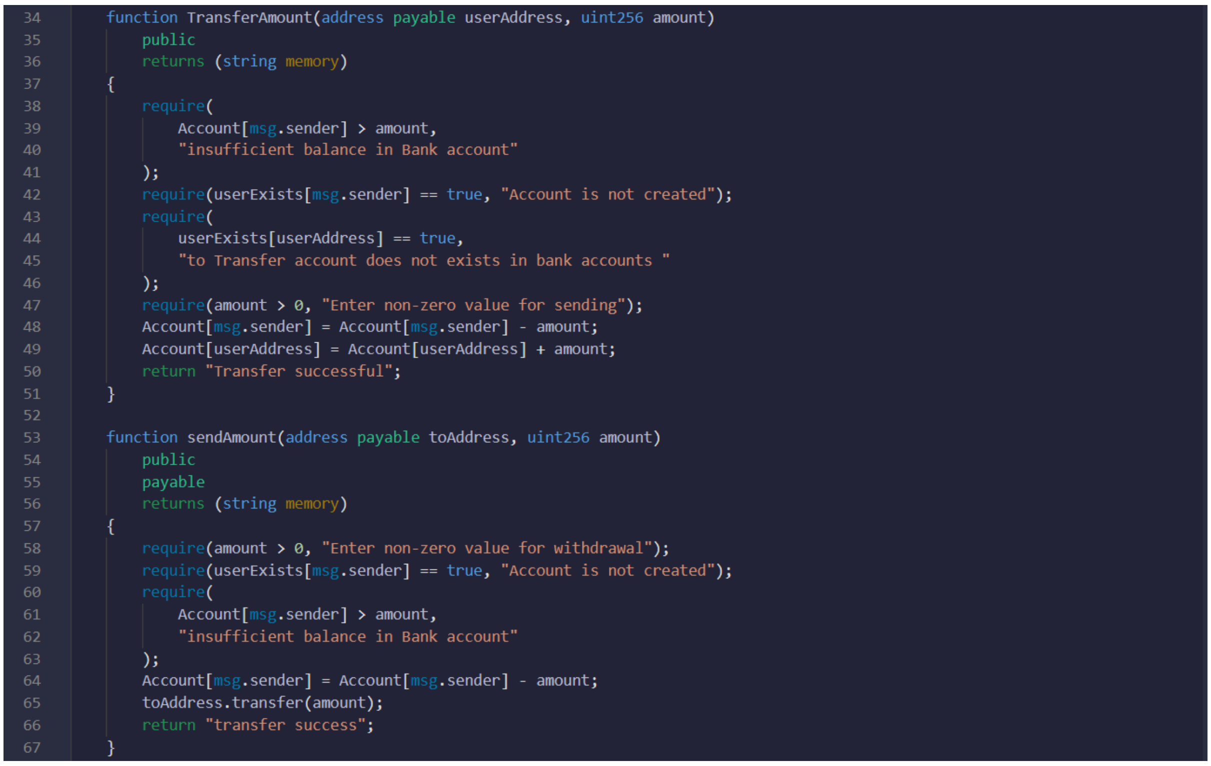
**Aim:** Create the Banking Application, Deploy it on Testnet through Metamask

## Source Code:

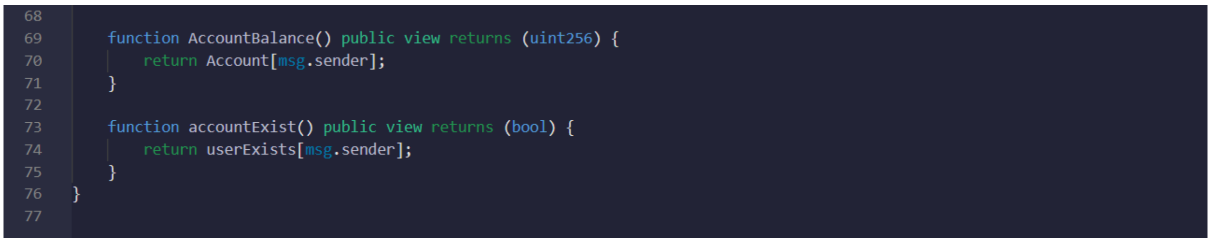
- ***Smart Contract*:**



```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract banking {
    mapping(address => uint256) public Account;
    mapping(address => bool) public userExists;

    function createAccount() public payable returns (string memory) {
        require(userExists[msg.sender] == false, "Account Already Exists");
        Account[msg.sender] = msg.value;
        userExists[msg.sender] = true;
        return "account created";
    }

    function deposit(uint256 amount) public payable returns (string memory) {
        require(userExists[msg.sender] == true, "Account is not created");
        require(amount > 0, "Value for deposit is Zero");
        Account[msg.sender] = Account[msg.sender] + amount;
        return "Deposited Succesfully";
    }

    function withdraw(uint256 amount) public payable returns (string memory) {
        require(
            Account[msg.sender] > amount,
            "Insufficeint balance in Bank account"
        );
        require(userExists[msg.sender] == true, "Account is not created");
        require(amount > 0, "Enter non-zero value for withdrawal");
        Account[msg.sender] = Account[msg.sender] - amount;
        msg.sender.transfer(amount);
        return "Withdrawal Succesful";
    }

```

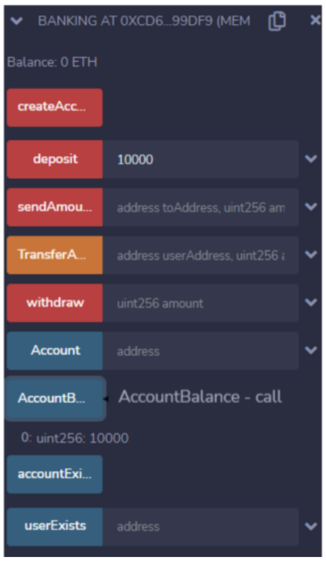


```
    function TransferAmount(address payable userAddress, uint256 amount)
        public
        returns (string memory)
    {
        require(
            Account[msg.sender] > amount,
            "insufficient balance in Bank account"
        );
        require(userExists[msg.sender] == true, "Account is not created");
        require(
            userExists[userAddress] == true,
            "to Transfer account does not exists in bank accounts "
        );
        require(amount > 0, "Enter non-zero value for sending");
        Account[msg.sender] = Account[msg.sender] - amount;
        Account[userAddress] = Account[userAddress] + amount;
        return "Transfer successful";
    }

    function sendAmount(address payable toAddress, uint256 amount)
        public
        payable
        returns (string memory)
    {
        require(amount > 0, "Enter non-zero value for withdrawal");
        require(userExists[msg.sender] == true, "Account is not created");
        require(
            Account[msg.sender] > amount,
            "insufficient balance in Bank account"
        );
        Account[msg.sender] = Account[msg.sender] - amount;
        toAddress.transfer(amount);
        return "transfer success";
    }
```



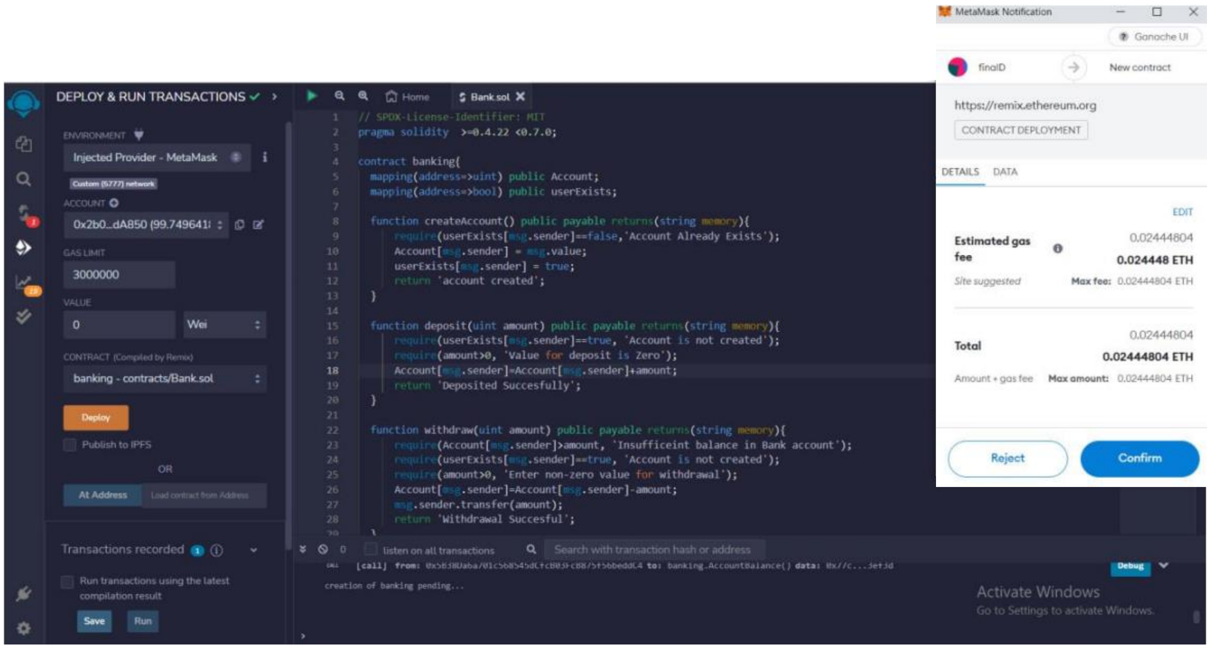
```

    function AccountBalance() public view returns (uint256) {
        return Account[msg.sender];
    }

    function accountExist() public view returns (bool) {
        return userExists[msg.sender];
    }
}

```

## Output:



```
BANKING AT 0XCD6...99DF9 (MEM
Balance: 0 ETH
createAcc...
deposit       10000
sendAmou...   address toAddress, uint256 am
TransferA...  address userAddress, uint256 a
withdraw      uint256 amount
Account       address
AccountB...   AccountBalance - call
0: uint256: 10000
accountExi...
userExists    address
```

## Deploying the smart contract in Ganache UI and Metamask:


DEPLOY & RUN TRANSACTIONS
Home
Bank.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 1
Save
Run
listen on all transactions
Search with transaction hash or address
Debug
Activate Windows
Go to Settings to activate Windows.
MetaMask Notification
DETAILS
DATA
EDIT
Reject
Confirm

ENVIRONMENT: Injected Provider - MetaMask
Custom (5777) network
ACCOUNT: 0x2b0...dA850 (99.749641
GAS LIMIT: 3000000
VALUE: 0 Wei
CONTRACT (Compiled by Remix): banking - contracts/Bank.sol

Run transactions using the latest compilation result

```
// SPDX-License-Identifier: MIT
pragma solidity  >=0.4.22 <0.7.0;

contract banking{
  mapping(address=>uint) public Account;
  mapping(address=>bool) public userExists;

  function createAccount() public payable returns(string memory){
      require(userExists[msg.sender]==false,'Account Already Exists');
      Account[msg.sender] = msg.value;
      userExists[msg.sender] = true;
      return 'account created';
  }

  function deposit(uint amount) public payable returns(string memory){
      require(userExists[msg.sender]==true, 'Account is not created');
      require(amount>0, 'Value for deposit is Zero');
      Account[msg.sender]=Account[msg.sender]+amount;
      return 'Deposited Succesfully';
  }

  function withdraw(uint amount) public payable returns(string memory){
      require(Account[msg.sender]>amount, 'Insufficeint balance in Bank account');
      require(userExists[msg.sender]==true, 'Account is not created');
      require(amount>0, 'Enter non-zero value for withdrawal');
      Account[msg.sender]=Account[msg.sender]-amount;
      msg.sender.transfer(amount);
      return 'Withdrawal Succesful';
  }
```

```
[call] from: 0x5838Daba701c568545dCfcB03FcB875f56beddC4 to: banking.AccountBalance() data: 0x77c...3ef3d
creation of banking pending...
```

Ganache UI
finalD → New contract
https://remix.ethereum.org
CONTRACT DEPLOYMENT

Estimated gas fee: 0.02444804 — **0.024448 ETH**
Site suggested — **Max fee:** 0.02444804 ETH

Total: 0.02444804 — **0.02444804 ETH**
Amount + gas fee — **Max amount:** 0.02444804 ETH