**User Identity Verification**

**Problem Statement**:

A user is trying to access your application by entering his private details. Develop a smart contract to verify the user’s identity before allowing him the access to your application.

**Steps to Perform:**

Step 1: Open Remix IDE

Step 2: Write the following contract in the Remix IDE:

pragma solidity ^0.4.17;

contract UserExample {

mapping(address => bool) user\_verified;

mapping(address => bytes32) user\_codes;

 mapping(bytes32 => address) to\_sign;

mapping(bytes32 => bool) signed;

 address owner;

 modifier onlyOwner() {

require(msg.sender == owner);

\_;

}

 function UserExample() public {

owner = msg.sender;

}

function addUser(address user, bytes32 hashed\_verification) public onlyOwner {

user\_verified[user] = false;

user\_codes[user] = hashed\_verification;

}

 function verify(bytes32 verification\_code) public {

if (user\_verified[msg.sender] == false &&

sha256(verification\_code) == user\_codes[msg.sender]) {

user\_verified[msg.sender] = true;

}

}

}

Step 3: Visit <https://www.myetherwallet.com/> and navigate to Contracts tab

Step 4: Click Deploy Contract

Step 5: Compile the solidity contract in Remix IDE

Step 6: Click on the details on the right-hand side of the Remix IDE, and use the object value of the byte code

Step 7: Copy the byte code, and paste it in the Byte Code field of the MyEtherWallet

Step 8: Select the Kovan network from the drop-down menu in the MyEtherWallet

Step 9: Create a wallet

Step 10: Visit [https://faucet.kovan.network/ to request the Kovan TestNet Community for test Ethers](https://faucet.kovan.network/)

Step 11: Unlock the wallet, and click Deploy after receiving the test network