

LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE
OF COMPUTER APPLICATION, MUMBAI

Blockchain & Solidity Program Lab Manual

**VIKRAM PATALBANSI,
DIVAKAR JHA**

Faculty, Master of Computer Application (M.C.A.)

Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application



BLOCKCHAIN & SOLIDITY PROGRAM LAB MANUAL

BLOCKCHAIN & SOLIDITY PROGRAM LAB MANUAL

LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE OF COMPUTER
APPLICATION, BANDRA, EAST MUMBAI-51

Author: Vikram Patalbansi
Co- Authors: Divakar Jha

*Faculty, Master of Computer Application (M.C.A.)
Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application*



First Edition, 2023

Copyright © Late Bhausaheb Hiray S.S. Trust's Institute Of Computer Application, Bandra (E), Mumbai-51, 2023

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the author, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, write to the publisher at the address below.

This book can be exported from India only by the publishers or by the authorized suppliers. Infringement of this condition of sale will lead to Civil and Criminal prosecution.

Paperback ISBN: 978-81-19221-67-7

eBook ISBN: 978-81-19221-64-6

WebPDF ISBN: 978-81-19221-66-0

Note: Due care and diligence has been taken while editing and printing the book; neither the author nor the publishers of the book hold any responsibility for any mistake that may have inadvertently crept in.

The publishers shall not be liable for any direct, consequential, or incidental damages arising out of the use of the book. In case of binding mistakes, misprints, missing pages, etc., the publishers' entire liability, and your exclusive remedy, is replacement of the book within one month of purchase by similar edition/reprint of the book.

Printed and bound in India by

16Leaves

2/579, Singaravelan Street

Chinna Neelankarai

Chennai - 600 041, India

info@16leaves.com

www.16Leaves.com

Call: 91-9940638999

CONTENTS

1 Blockchain's Block creation using JavaScript	1
2 Steps for Blockchain program compiling and running using Truffle DApp program using Metamask	5
3 MetaMask Configuration Steps and Transaction	21
4 Blockchain Cryptography Program using Java	33
5 Blockchain Public Key Cryptography	35
6 Blockchain cryptography using Message Digest algorithm in Java	37
7 Program using Solidity to check Balance	39
8 Contract using Structure in Solidity Code	51
9 Ethereum Installation	63
10 Program on Election DApp	75
11 The use of GANACHE Truffle Suite to Deploy a Smart Contract in Solidity (Blockchain)	83
12 How to use MetaMask to Deploy a Smart contract in Solidity (Blockchain)?	91
13 How to Develop Ethereum Smart Contract with Truffle and Ganache	97
14 Blockchain creation program using Java	105
15 Simple Solidity Program in Ethereum	115
16 Ethereum Development program to check balance	125
17 Ethereum program on Solidity to check account details	137
18 Working with Smart Contracts in Ethereum	149
19 Syntaxes in Solidity Programs	171
20 Simple Calculator Solidity Programs	203
21 Creating the Smart Contract	209

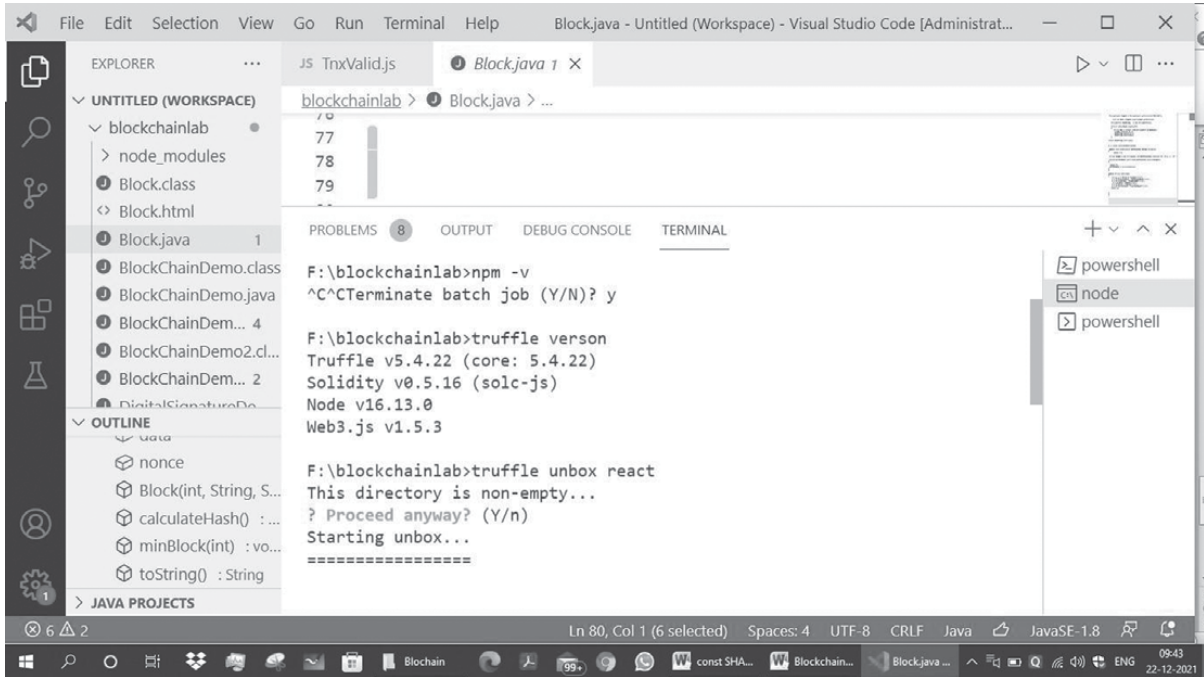
22 Smart Contract using Web3.js	223
23 DApp on Election	227
24 Case Study on Blockchain	251
25 Annexure-I	277

Prog. 1 Blockchain's Block creation using JavaScript

```
const SHA256 = require("crypto-js/sha256"); class Block
{
  constructor(index,timestamp,data,previoushash="")
  {
    this.index = index;
    this.timestamp = timestamp;
    this.data = data;
    this.previoushash = previoushash;
    this.hash = this.calculateHash();
  }
  calculateHash()
  {
    return SHA256(this.index+this.timestamp+this.previoushash+JSON.stringify(this.data)).toString();
  }
} //close block class class Blockchain
{
  constructor(index,timestamp,data,previoushash="")
  {
    this.index = index;
    this.timestamp = timestamp;
    this.data = data;
    this.previoushash = previoushash;
    this.chain =[this.createGenesisBlock()];
  }
  createGenesisBlock()
  {
    return new Block(0,"23/11/2021","This is First Program of Blockchain creation","0");
  }
  addBlock(newBlock)
  {
    newBlock.previoushash = this.getLatestBlock().hash;
    newBlock.hash = newBlock.calculateHash();
    this.chain.push(newBlock);
  }
  getLatestBlock()
  {
    return this.chain[this.chain.length-1];
  }
} // close blockchain class
```

```
let block1 = new Block(1,"22/11/2021","Data1","0");
let block2 = new Block(2,"21/11/2021","Second Block","");
let block3 = new Block(3,"14/05/2021","Third Block","");
let myBlockchain = new Blockchain(); myBlockchain.addBlock(block1);
myBlockchain.addBlock(block2); myBlockchain.addBlock(block3);
console.log(JSON.stringify(myBlockchain,null,4));
```

output:



```
F:\blockchainlab>npm -v
^C^CTerminate batch job (Y/N)? y

F:\blockchainlab>truffle version
Truffle v5.4.22 (core: 5.4.22)
Solidity v0.5.16 (solc-js)
Node v16.13.0
Web3.js v1.5.3

F:\blockchainlab>truffle unbox react
This directory is non-empty...
? Proceed anyway? (Y/n)
Starting unbox...
=====
```

```

File Edit Selection View Go Run Terminal Help • jsBlockchainex.js - Untitled (Workspace) - Visual Studio Code [Adminis...
EXPLORER
UNTITLED (WORKSPACE)
  blockchainlab > JS jsBlockchainex.js > ...
  JS Hello.js
  JS jsBlockchain.js
  JS jsBlockchainex.js
  MessageDigest2.class
  MessageDigest... 1
  MessageDigestEncD...
  MessageDigestEncD...
  package-lock.json
  package.json
  PrivateKeyDemo.class
OUTLINE
  SHA256
  Block
    constructor
    calculateHash
  Blockchain
  JAVA PROJECTS
  6 2

blockchainlab > JS jsBlockchainex.js > ...
49
50 let myBlockchain = new Blockchain();
51 myBlockchain.addBlock(block1);
52 myBlockchain.addBlock(block2);

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL
F:\blockchainlab>
F:\blockchainlab>node jsBlockchainex.js
{
  "previoushash": "",
  "chain": [
    {
      "index": 0,
      "timestamp": "23/11/2021",
      "data": "This is First Program of Blockchain creation",
      "previoushash": "0",
      "hash": "ddad1af2b2f82b4a13c7a701909851471bb1879bbe2ed8f854ff
b7ad67277471"
    }
  ]
}
Ln 51, Col 31 Spaces: 4 UTF-8 CRLF JavaScript 09:46 22-12-2021

```

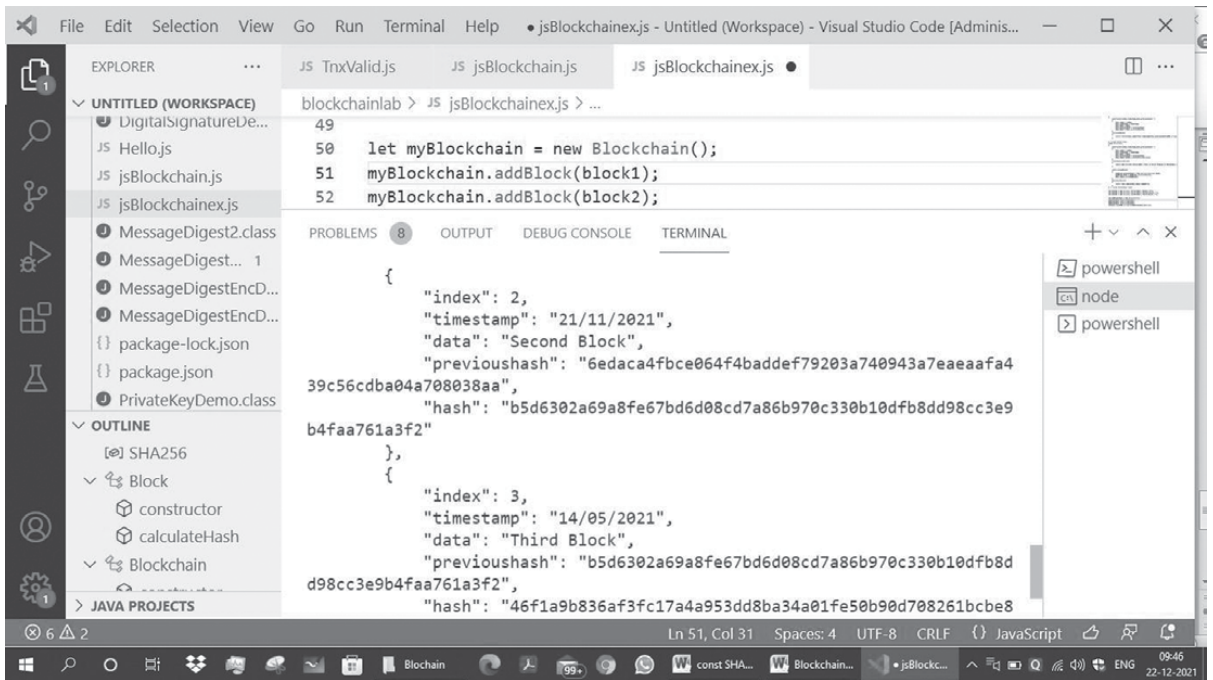
```

File Edit Selection View Go Run Terminal Help • jsBlockchainex.js - Untitled (Workspace) - Visual Studio Code [Adminis...
EXPLORER
UNTITLED (WORKSPACE)
  blockchainlab > JS jsBlockchainex.js > ...
  JS Hello.js
  JS jsBlockchain.js
  JS jsBlockchainex.js
  MessageDigest2.class
  MessageDigest... 1
  MessageDigestEncD...
  MessageDigestEncD...
  package-lock.json
  package.json
  PrivateKeyDemo.class
OUTLINE
  SHA256
  Block
    constructor
    calculateHash
  Blockchain
  JAVA PROJECTS
  6 2

blockchainlab > JS jsBlockchainex.js > ...
49
50 let myBlockchain = new Blockchain();
51 myBlockchain.addBlock(block1);
52 myBlockchain.addBlock(block2);

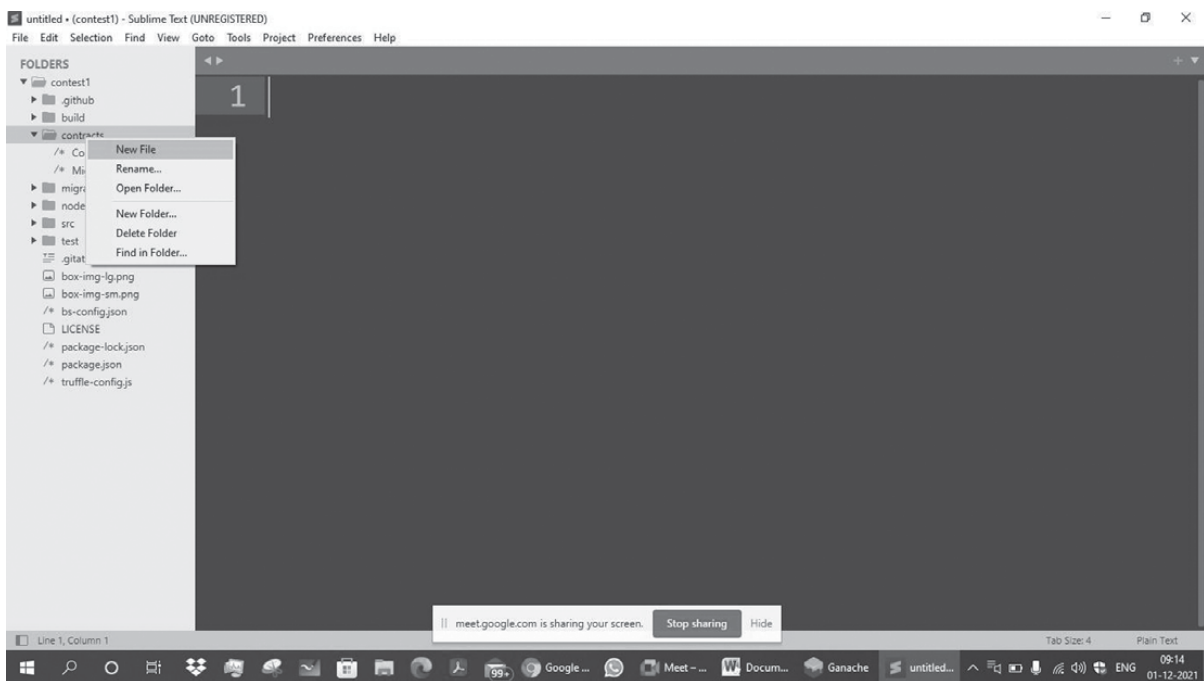
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL
b7ad67277471"
  },
  {
    "index": 1,
    "timestamp": "22/11/2021",
    "data": "Data1",
    "previoushash": "ddad1af2b2f82b4a13c7a701909851471bb1879bbe2e
d8f854fffb7ad67277471",
    "hash": "6edaca4fbce064f4baddef79203a740943a7eaeafa439c56cdb
a04a708038aa"
  },
  {
    "index": 2,
    "timestamp": "21/11/2021",
    "data": "Second Block",
    "previoushash": "6edaca4fbce064f4baddef79203a740943a7eaeafa4
Ln 51, Col 31 Spaces: 4 UTF-8 CRLF JavaScript 09:46 22-12-2021

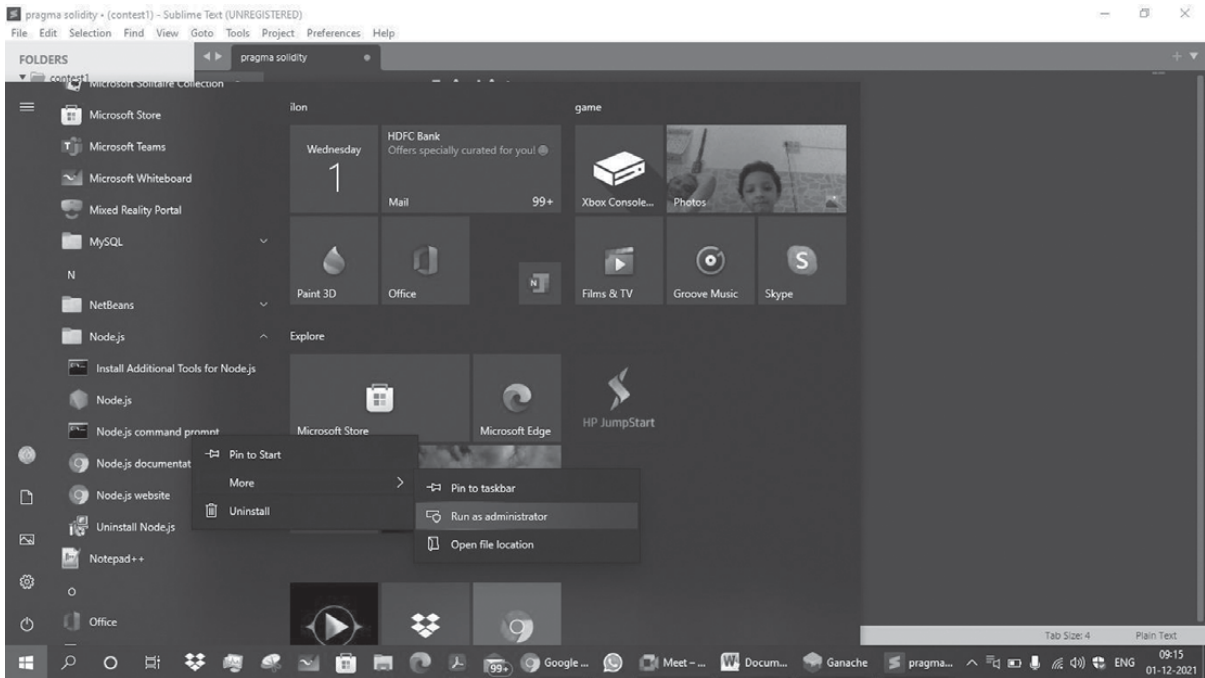
```



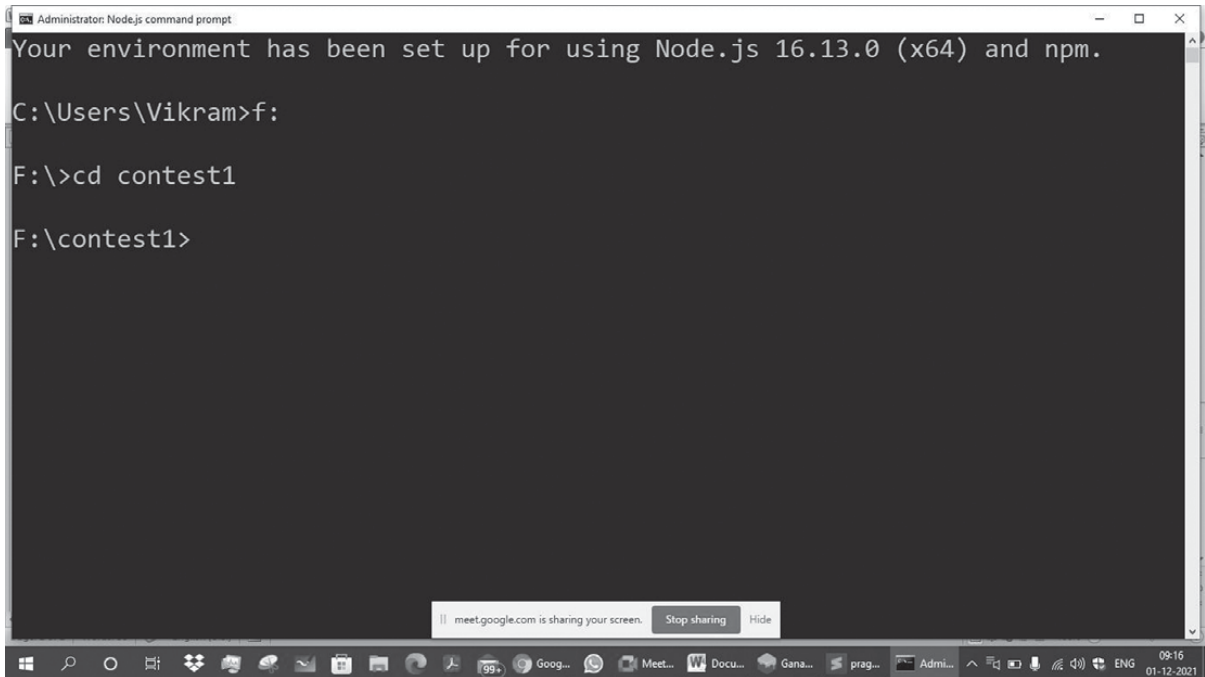
Prog. 2 Steps for Blockchain program compiling and running using Truffle DApp program using Metamask

- 1) Open the Ganach
- 2) Open Sublimetext3 editor
- 3) create the folder contest1
- 4) install the truffle in the directory F:/>contest1>npm install truffle -g
- 5) Make the contract Election.sol using Solidity Program

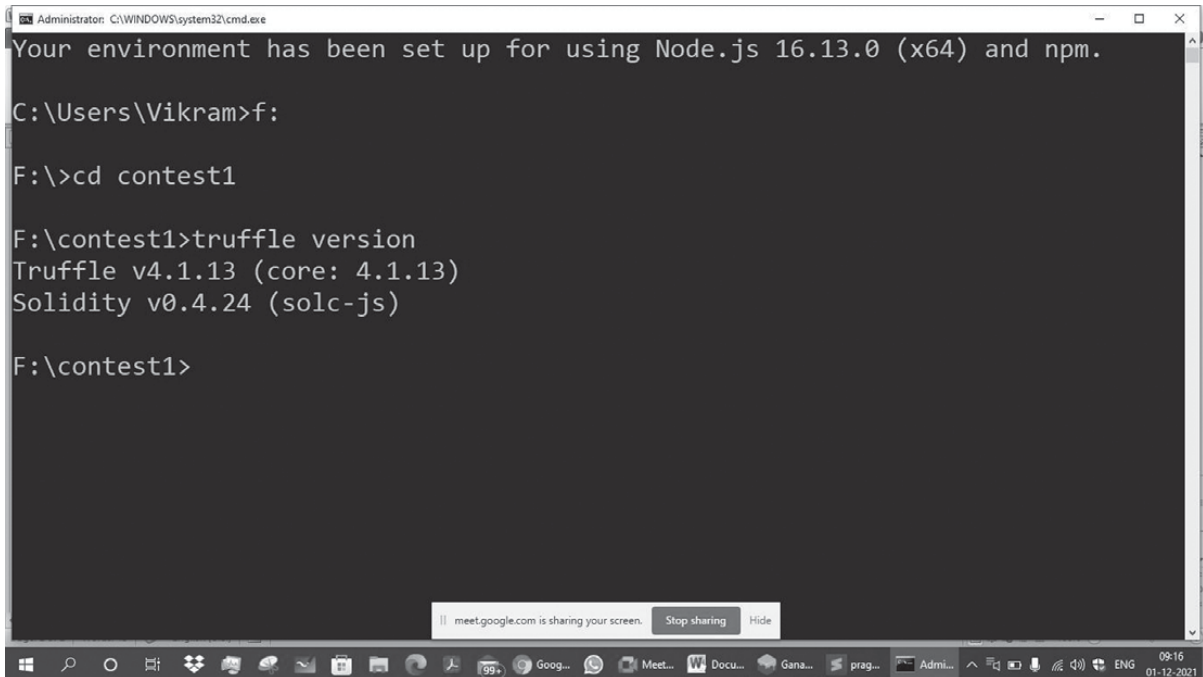




7) f:



8) Check the truffle version



```
Administrator: C:\WINDOWS\system32\cmd.exe
Your environment has been set up for using Node.js 16.13.0 (x64) and npm.

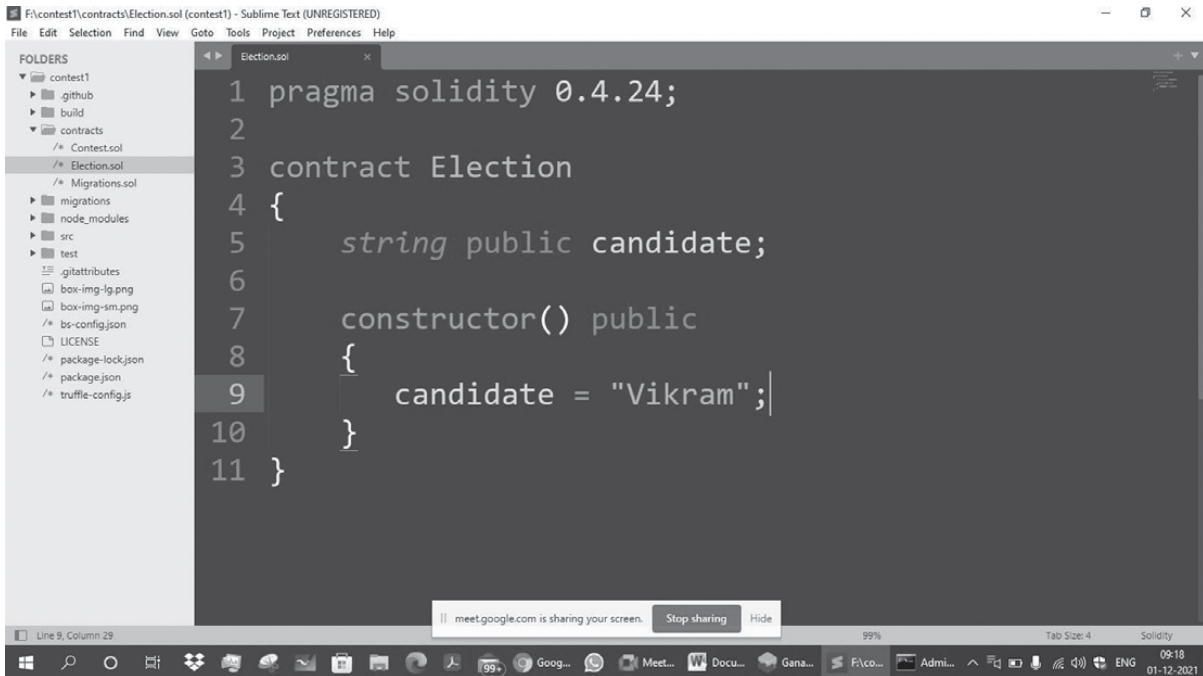
C:\Users\Vikram>f:

F:\>cd contest1

F:\contest1>truffle version
Truffle v4.1.13 (core: 4.1.13)
Solidity v0.4.24 (solc-js)

F:\contest1>
```

9) Write Solidity Program

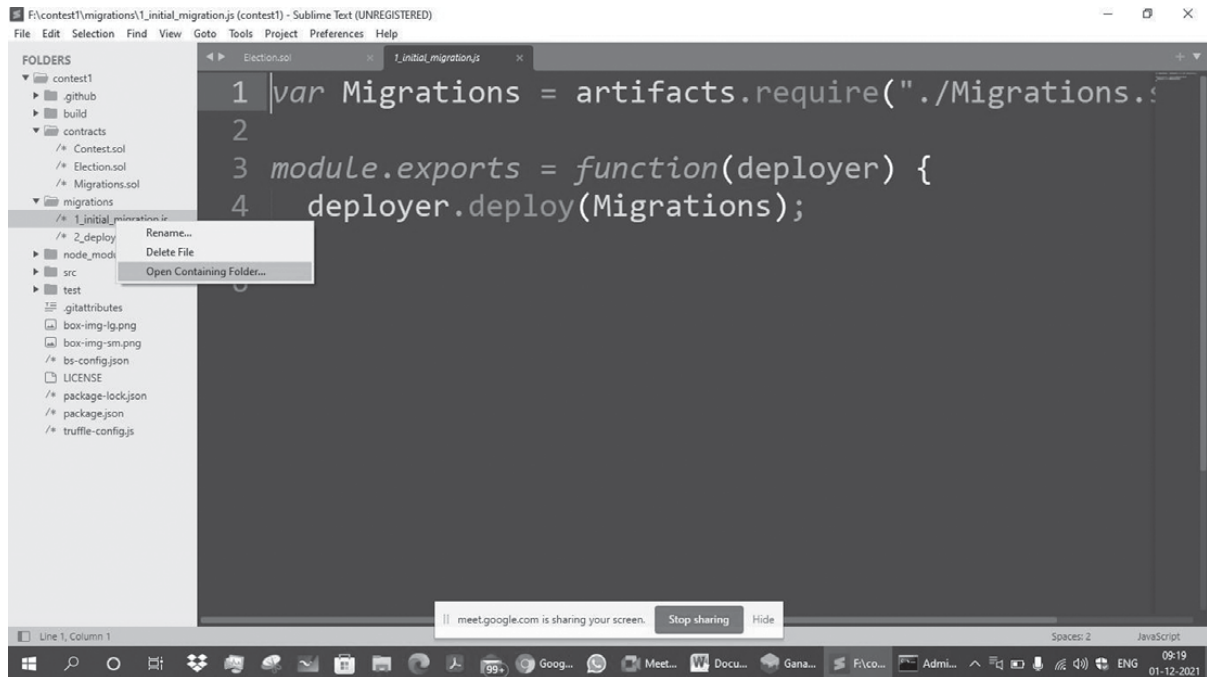


```
F:\contest1\contracts\Election.sol (contest1) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ contest1
  └─ .github
  └─ build
  └─ contracts
     └─ Contest.sol
     └─ Election.sol
     └─ Migrations.sol
  └─ migrations
  └─ node_modules
  └─ src
  └─ test
  └─ gitattributes
  └─ box-img-1g.png
  └─ box-img-sm.png
  └─ bs-config.json
  └─ LICENSE
  └─ package-lock.json
  └─ package.json
  └─ truffle-config.js

Election.sol
1 pragma solidity 0.4.24;
2
3 contract Election
4 {
5     string public candidate;
6
7     constructor() public
8     {
9         candidate = "Vikram";
10    }
11 }
```

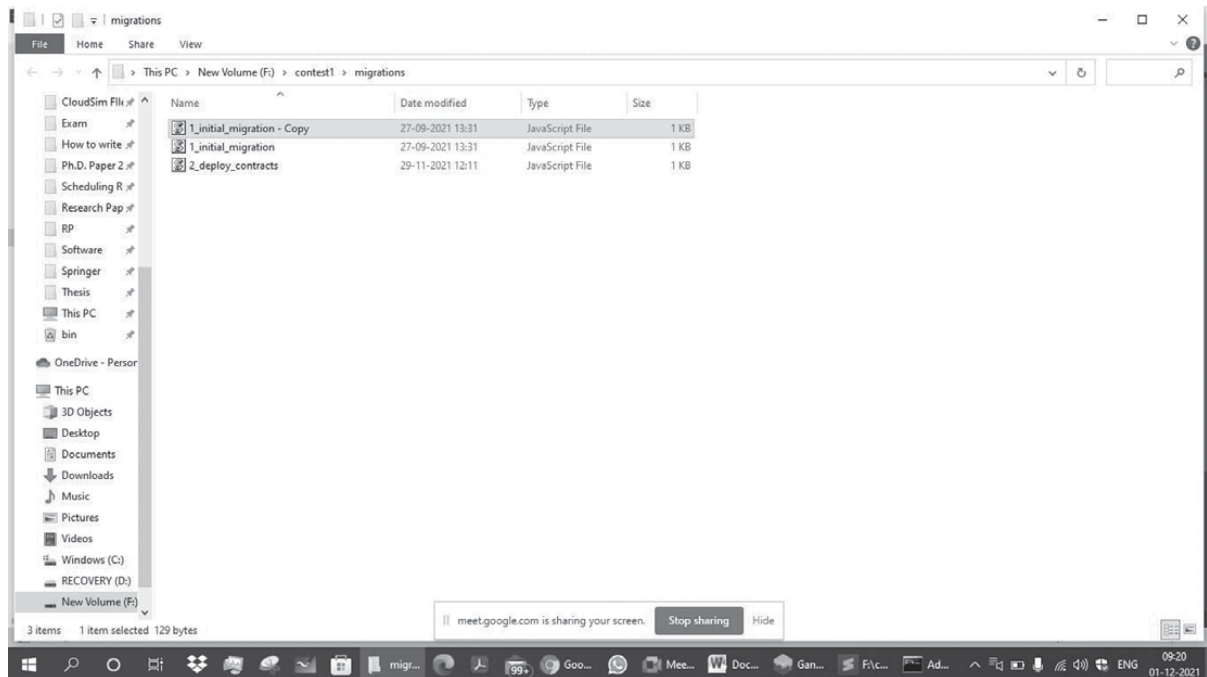
10) Configure the deployment files in migration folder

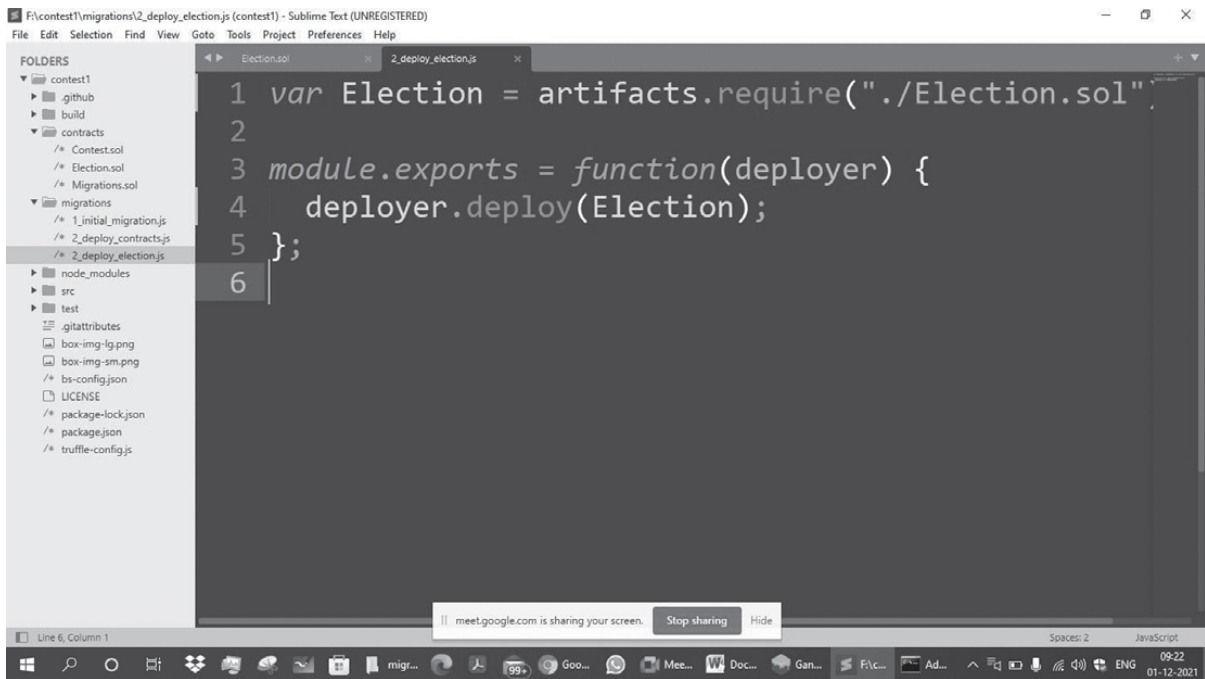
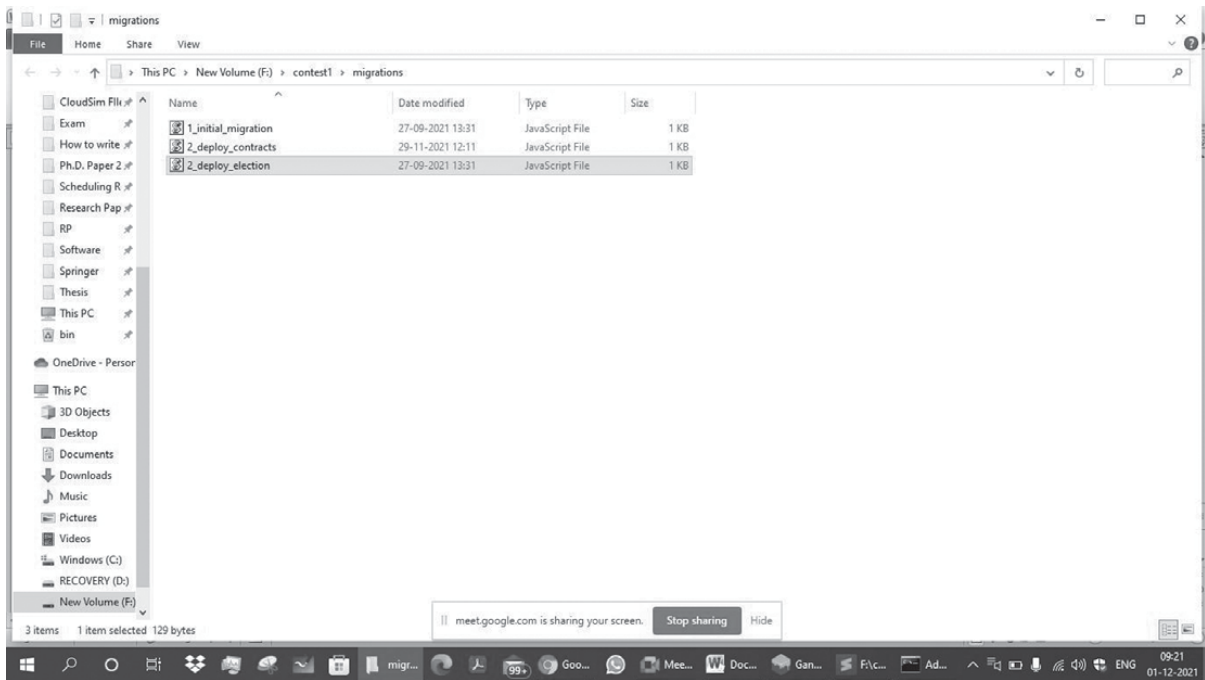


The screenshot shows the Sublime Text editor with the file `1_initial_migration.js` open. The code is as follows:

```
1 var Migrations = artifacts.require("./Migrations.sol");
2
3 module.exports = function(deployer) {
4   deployer.deploy(Migrations);
5 }
```

The file explorer on the left shows the project structure, with the `migrations` folder selected. A context menu is open over `1_initial_migration.js`, showing options like `Rename...`, `Delete File`, and `Open Containing Folder...`.





12) Compile the Contract Program

```
Administrator: C:\WINDOWS\system32\cmd.exe
Your environment has been set up for using Node.js 16.13.0 (x64) and npm.

C:\Users\Vikram>f:

F:\>cd contest1

F:\contest1>truffle version
Truffle v4.1.13 (core: 4.1.13)
Solidity v0.4.24 (solc-js)

F:\contest1>truffle migrate
```

```
Administrator: C:\WINDOWS\system32\cmd.exe
Saving successful migration to network...
... 0xb77ed593bef43c89677f6481a184bb0837b713a4b41eebb80e2d03b378111e68
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Replacing Contest...
  ... 0x6af7e376a6ed5b6e2542c4349d63ff78d461a21800463e4a3f5b72ad085d5bb7
  Contest: 0x2d49aae9ca9683d36a47da64f94c9a35de31eb00
Saving successful migration to network...
... 0x78043cc74bd7f2ef09d09a335127c014f553d468c058b54bc9e32826239eae8
Saving artifacts...
Running migration: 2_deploy_election.js
  Deploying Election...
  ... 0xe1a2d9e3ce181dbe34093a133e3a034b9f18fe034c18145fc4843bf90da033b2
  Election: 0x5882bece78a64453f2c3b05609635fc039f8a6e7
Saving successful migration to network...
... 0xb154ea217b3c44df78000c7323f875b56eb06d77bbd11e0161b4d536007ccd
Saving artifacts...

F:\contest1>
```

13) Deploy the contract on blockchain network i.e. Ganache localhost blockchain

```
Administrator: C:\WINDOWS\system32\cmd.exe
Saving artifacts...
Running migration: 2_deploy_election.js
  Deploying Election...
  ... 0xe1a2d9e3ce181dbe34093a133e3a034b9f18fe034c18145fc4843bf90da033b2
  Election: 0x5882bece78a64453f2c3b05609635fc039f8a6e7
Saving successful migration to network...
  ... 0xb154ea217b3c44df780000c7323f875b56eb06d77bbd11e0161b4d536007cced
Saving artifacts...

F:\contest1>truffle console
```

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
Replacing Contest...
  ... 0x6af7e376a6ed5b6e2542c4349d63ff78d461a21800463e4a3f5b72ad085d5bb7
  Contest: 0x2d49aae9ca9683d36a47da64f94c9a35de31eb00
Saving successful migration to network...
  ... 0x78043cc74bd7f2ef09d09a335127c014f553d468c058b54bc9e32826239eae8
Saving artifacts...
Running migration: 2_deploy_election.js
  Deploying Election...
  ... 0xe1a2d9e3ce181dbe34093a133e3a034b9f18fe034c18145fc4843bf90da033b2
  Election: 0x5882bece78a64453f2c3b05609635fc039f8a6e7
Saving successful migration to network...
  ... 0xb154ea217b3c44df780000c7323f875b56eb06d77bbd11e0161b4d536007cced
Saving artifacts...

truffle(development)> Election.deployed().then(function(instance){ app = insta
truffle(development)>
undefined
```

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
  request: [Function: bound ],
  estimateGas: [Function (anonymous)]
}
truffle(development)> app.candidate()
'Vikram'
```

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
  request: [Function: bound ],
  estimateGas: [Function (anonymous)]
}
truffle(development)>
'0x5882bece78a64453f2c3b05609635fc039f8a6e7'
```

```
1 pragma solidity 0.4.24;
2
3 contract Election
4 {
5     struct Candidate
6     {
7         uint id;
8         string name;
9         uint voteCount;
10    } // this is store candidate information
11
12    // Fetch candidate information using
13    // reference variable
14    mapping(uint => Candidate) public candidate;
```

```
pragma solidity 0.4.24;
```

```
contract Election
```

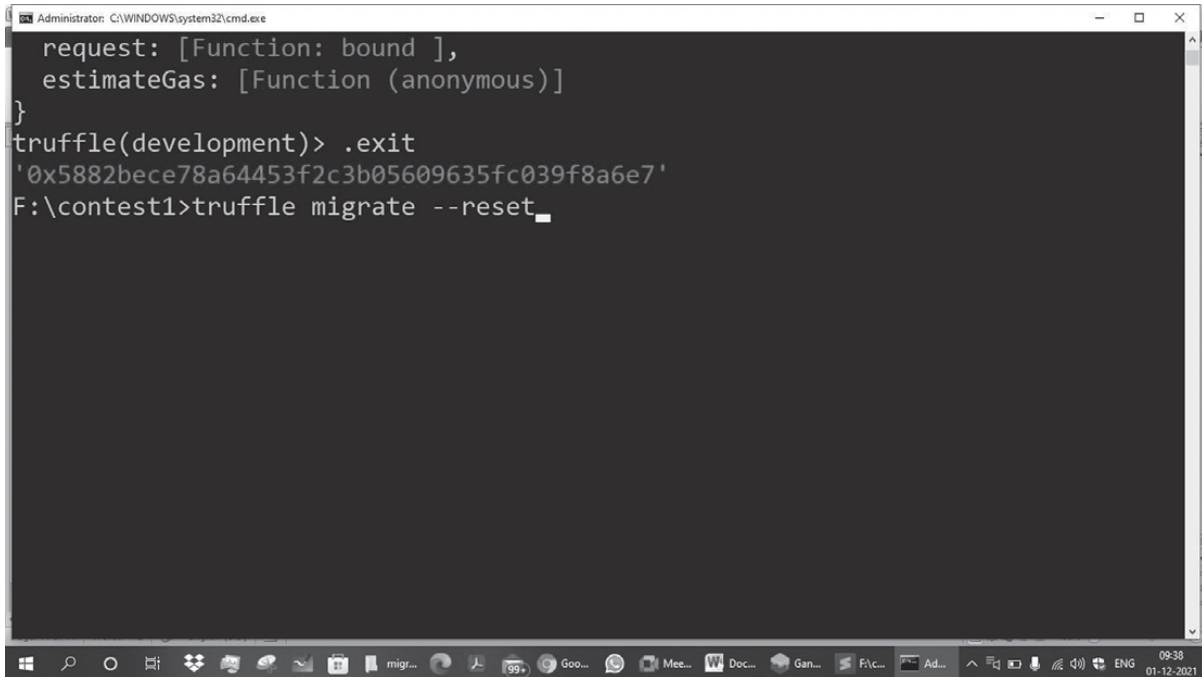
```
{
    struct Candidate
    {
        uint id;
        string name;
        uint voteCount;
    } // this is store candidate information

    // Fetch candidate information using
    // reference variable
    mapping(uint => Candidate) public candidate;
    //variable for vote counting
    uint public candidateCount;

    //constructor
    function Election() public
    {
        addCandidate("Sourabh");
        addCandidate("Neeta");
    }
}
```

```
// define addcandidate function
function addCandidate(string memory _name)private
{
    candidateCount++;
    candidate[candidateCount] = Candidate(candidateCount,_name,0);
}
}
```

14) Recompile the same Election contract



```
Administrator: C:\WINDOWS\system32\cmd.exe
request: [Function: bound ],
estimateGas: [Function (anonymous)]
}
truffle(development)> .exit
'0x5882bece78a64453f2c3b05609635fc039f8a6e7'
F:\contest1>truffle migrate --reset
```

```
Administrator: C:\WINDOWS\system32\cmd.exe
Using network 'development'.

Running migration: 1_initial_migration.js
  Replacing Migrations...
  ... 0x103c657319e1a0cddfadbdf92f387fbf3b1bfe2563d3c8bf82453f541dde600
  Migrations: 0xd26f89879fc89996f3626a2686515b30f43072eb
Saving successful migration to network...
  ... 0x4fcef3a1aa4dbf0b4f764e15aa4dd0caf2a084375a30443975d30842a009d471
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Replacing Contest...
  ... 0xe0c1a01e266c925165949980f8c52c820fd0e56b6e193ec8476fc0fcaca042b9
  Contest: 0x84fb6ecb0a8ee60930e899dce9b0b7ce903b0cbf
Saving successful migration to network...
  ... 0x611435eccc9fa00d918e8edab53ffdaecb056f14f29a6aada3a7824e49004478
Saving artifacts...
Running migration: 2_deploy_election.js
  Replacing Election...
  ... 0x7a7a0e50521e03077ea071b47e218ea0c18ba01473746e32bc67d23170a853ed
```

15) Deploy the Contract over the Ethereum Ganach

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli_bundled.js" console
  Contest: 0x84fb6ecb0a8ee60930e899dce9b0b7ce903b0cbf
Saving successful migration to network...
  ... 0x611435eccc9fa00d918e8edab53ffdaecb056f14f29a6aada3a7824e49004478
Saving artifacts...
Running migration: 2_deploy_election.js
  Replacing Election...
  ... 0x7a7a0e50521e03077ea071b47e218ea0c18ba01473746e32bc67d23170a853ed
  Election: 0x5934ffe6e51c741ed70b325d9ebef6a79674b497
Saving successful migration to network...
  ... 0x7a8e2101d98c64487b44afc0253fc91daddc3bd330b35eb6cea5924cabd7025b
Saving artifacts...

truffle(development)> Election.deployed().then(function(i){ app = i})
```

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
... 0x7a7a0e50521e03077ea071b47e218ea0c18ba01473746e32bc67d23170a853ed
Election: 0x5934ffe6e51c741ed70b325d9ebef6a79674b497
Saving successful migration to network...
... 0x7a8e2101d98c64487b44afc0253fc91daddc3bd330b35eb6cea5924cabd7025b
Saving artifacts...

truffle(development)> app.candidate(1)
[undefined
  BigNumber { s: 1, e: 0, c: [ 1 ] },
  'Sourabh',
  BigNumber { s: 1, e: 0, c: [ 0 ] }
]
```

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
... 0x7a7a0e50521e03077ea071b47e218ea0c18ba01473746e32bc67d23170a853ed
Election: 0x5934ffe6e51c741ed70b325d9ebef6a79674b497
Saving successful migration to network...
... 0x7a8e2101d98c64487b44afc0253fc91daddc3bd330b35eb6cea5924cabd7025b
Saving artifacts...

truffle(development)> app.candidate(2)
[undefined
  BigNumber { s: 1, e: 0, c: [ 2 ] },
  'Neeta',
  BigNumber { s: 1, e: 0, c: [ 0 ] }
]
truffle(development)>
```


16) Fetching candidate

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
'0x99381135f7f11e23132852be8bbbe491e42574dd',
'0x17921e1c0e6d9e1bea0c4524f36c8f54ba190e2f'
]
truffle(development)> web3.eth.accounts
```

```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
'0x99381135f7f11e23132852be8bbbe491e42574dd',
'0x17921e1c0e6d9e1bea0c4524f36c8f54ba190e2f'
]
truffle(development)> 
[
'0x56403fa625784b0c459a35ecc6469f67afb5a0b5',
'0x31e489dac46e9f1cb2cf73b2b401cbac2f879b65',
'0xbac19d44e4659c31f61f4f63699f0e3f28074999',
'0x1a9b2628957b15b1a78b3bc50438ba42554100aa',
'0x4ce61e0a1f478234b64f46357424d7b0b6145804',
'0x6ad6a00b1c275baec3091ecc0a8bc38a21d75ebe',
'0x877d120552525ef6835e196183d1e71dc0c27d99',
'0xc0e570a46cc3555400e96c4727d4b9175e4afccc',
'0x99381135f7f11e23132852be8bbbe491e42574dd',
'0x17921e1c0e6d9e1bea0c4524f36c8f54ba190e2f'
]
```

17) Fetch Individual Ganache node address

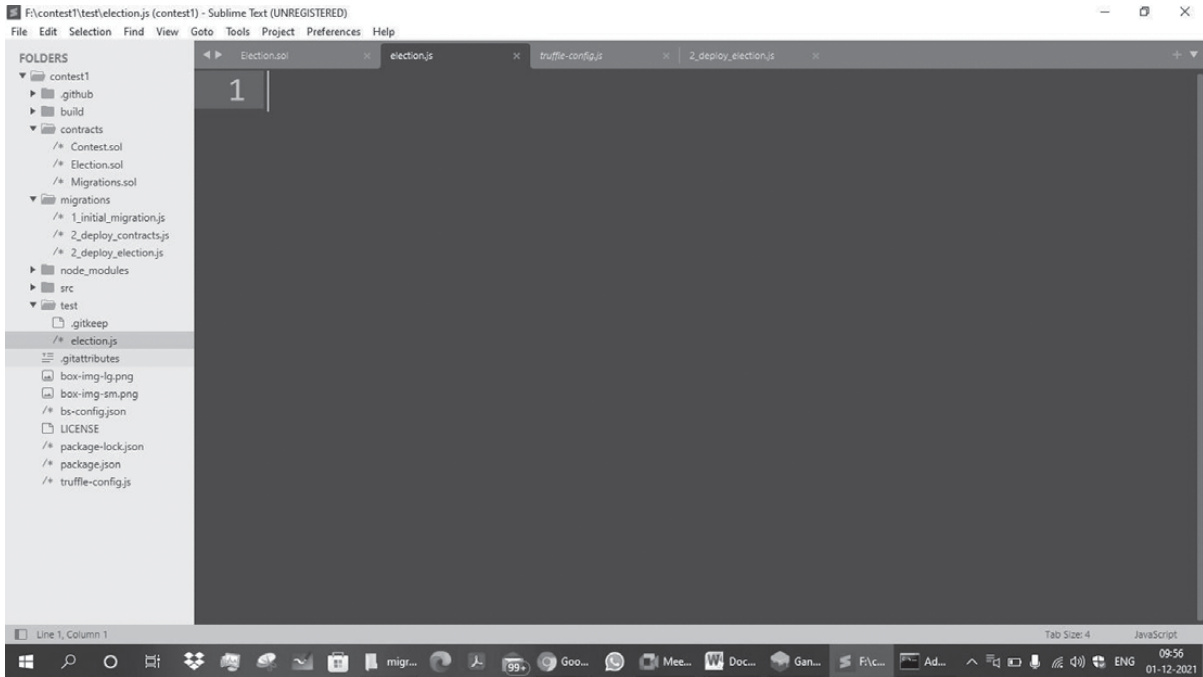
```
Administrator: C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\Vikram\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.js" console
'0x4ce61e0a1f478234b64f46357424d7b0b6145804',
'0x6ad6a00b1c275baec3091ecc0a8bc38a21d75ebe',
'0x877d120552525ef6835e196183d1e71dc0c27d99',
'0xc0e570a46cc3555400e96c4727d4b9175e4afccc',
'0x99381135f7f11e23132852be8bbbe491e42574dd',
'0x17921e1c0e6d9e1bea0c4524f36c8f54ba190e2f'
]
truffle(development)> web3.eth.accounts[4]
'0x4ce61e0a1f478234b64f46357424d7b0b6145804'
truffle(development)>
'0x31e489dac46e9f1cb2cf73b2b401cbac2f879b65',
'0xbac19d44e4659c31f61f4f63699f0e3f28074999',
'0x1a9b2628957b15b1a78b3bc50438ba42554100aa',
'0x4ce61e0a1f478234b64f46357424d7b0b6145804',
'0x6ad6a00b1c275baec3091ecc0a8bc38a21d75ebe',
'0x877d120552525ef6835e196183d1e71dc0c27d99',
'0xc0e570a46cc3555400e96c4727d4b9175e4afccc',
'0x99381135f7f11e23132852be8bbbe491e42574dd',
'0x17921e1c0e6d9e1bea0c4524f36c8f54ba190e2f'
```

18) Open the two library files to deploy contract on Ethereum

» a) <https://mochajs.org>

» b) www.chaijs.com

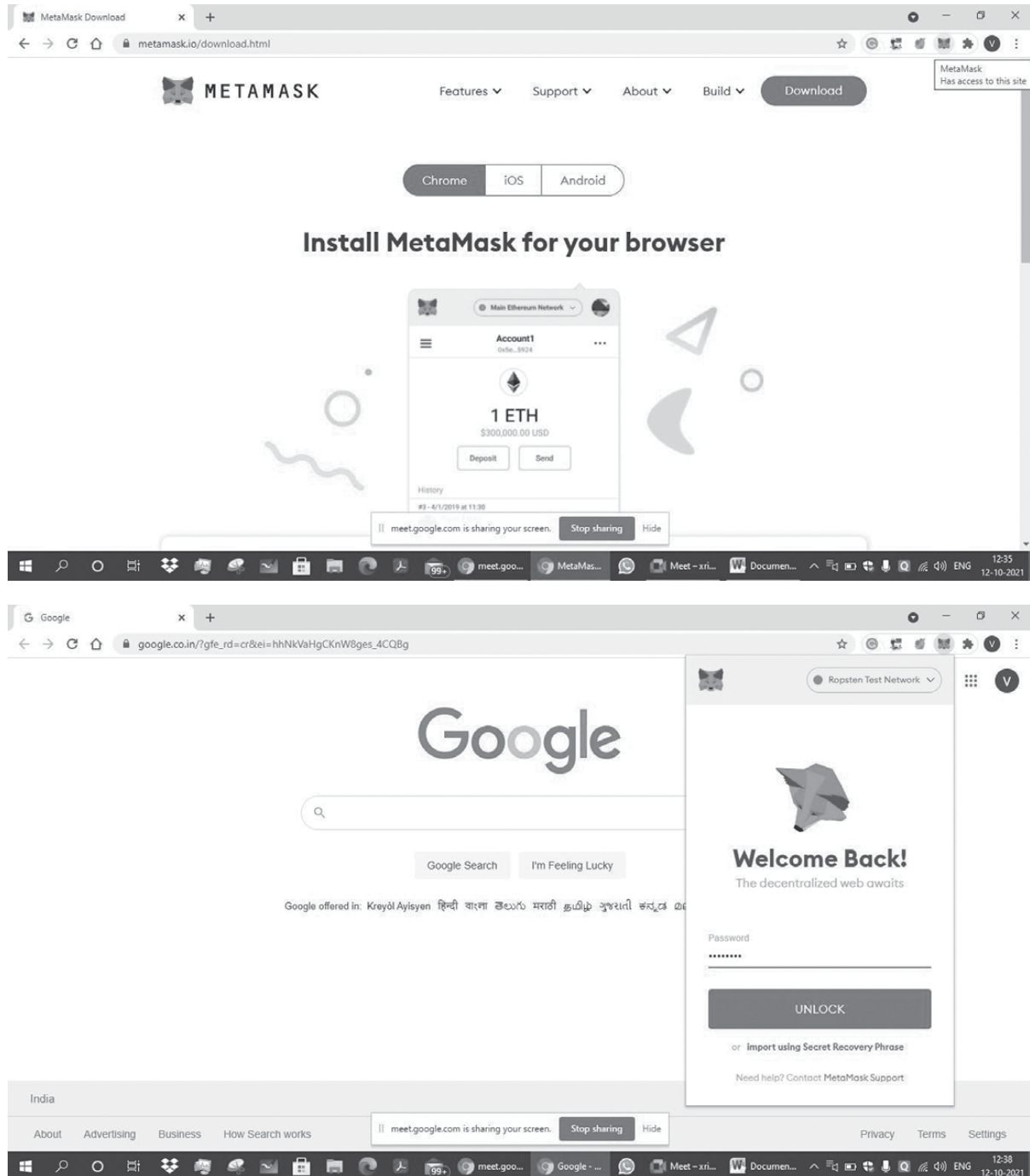
Here Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing.. Assertion Library: Chai – Chai is a BDD/TDD assertion library for[node](http://nodejs.org) and browser that can be paired with it In the Test folder we have to create election.js file in sublimetext3

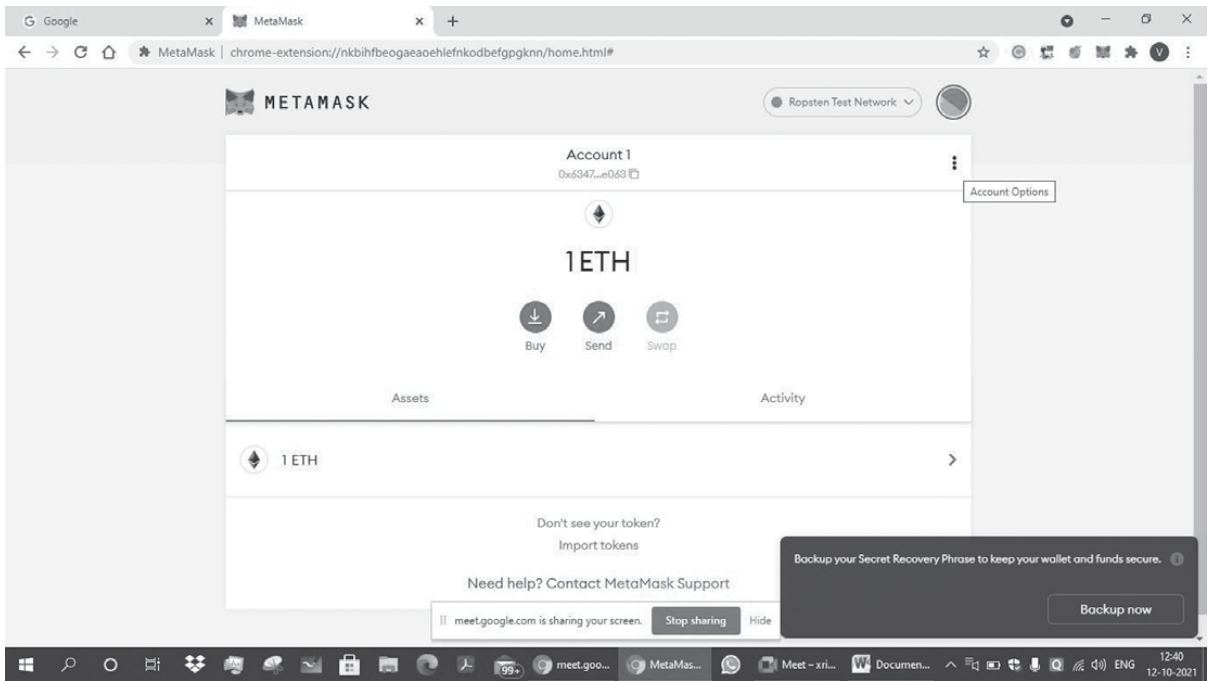
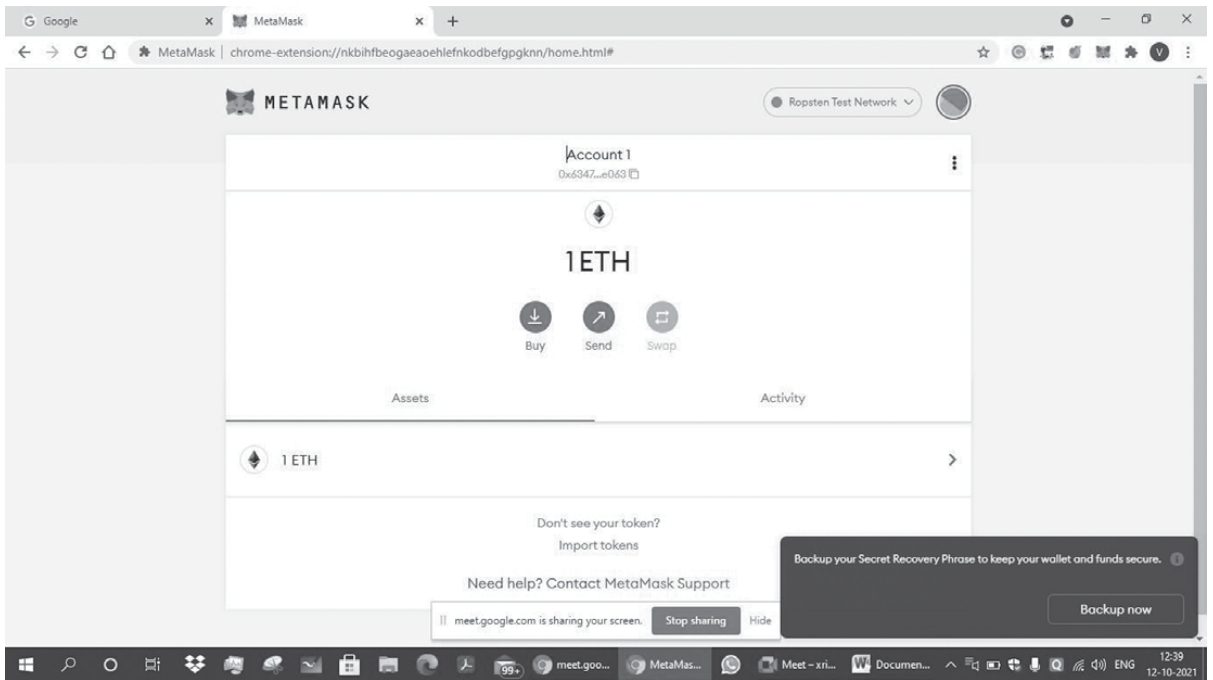


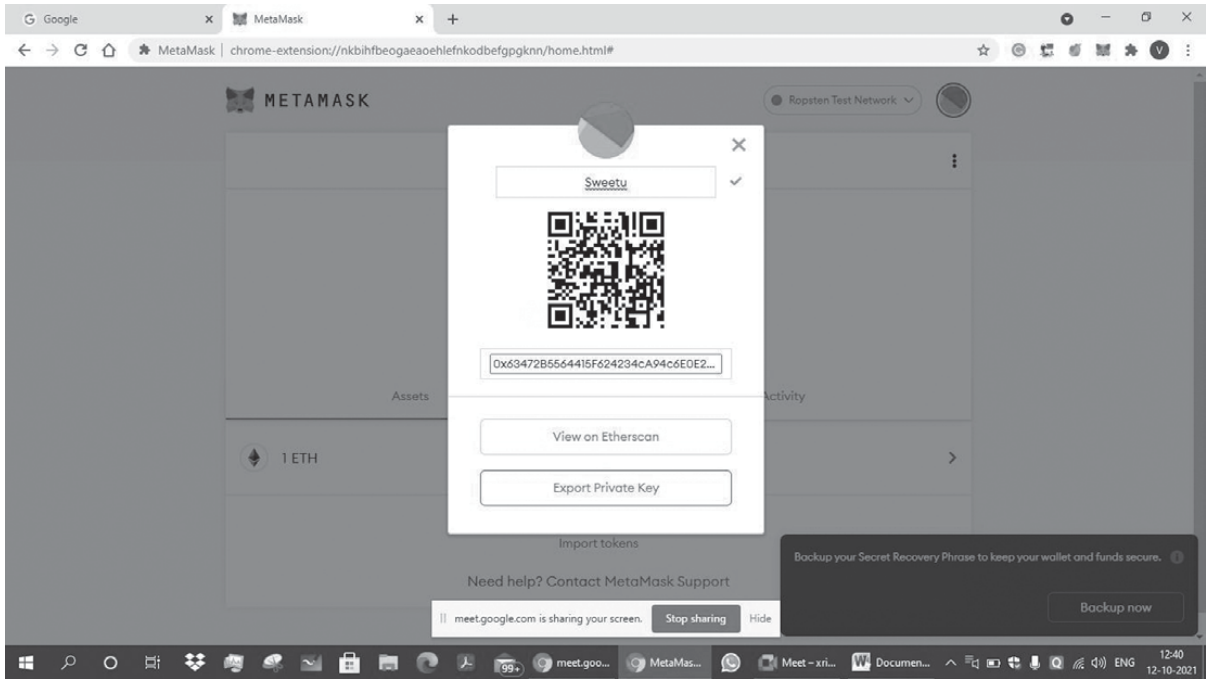
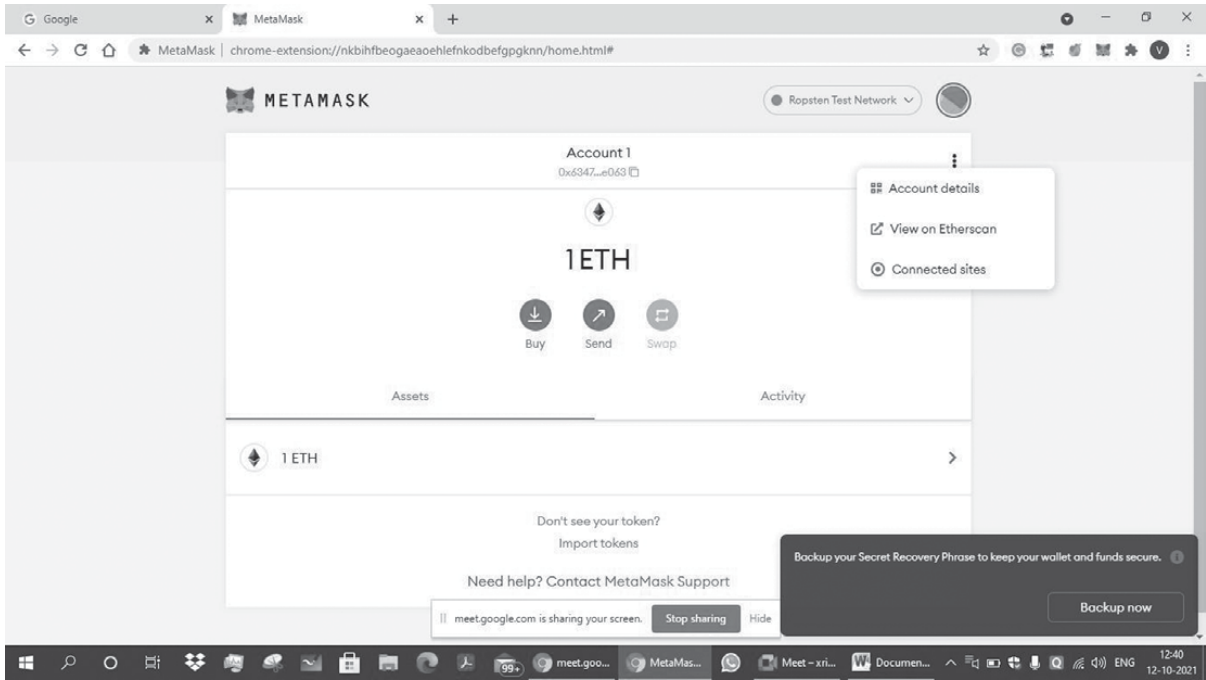
```
const Election = artifacts.require("./Election.sol");  
require('chai').use(require('chai-as-promised')).should();
```

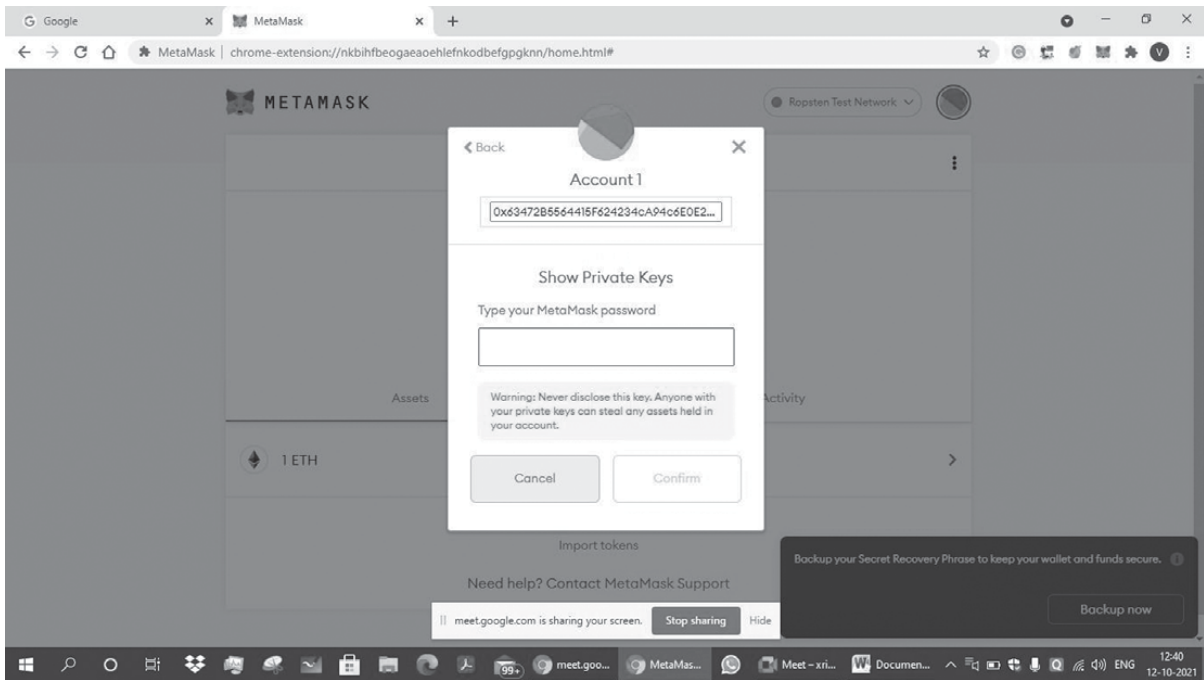
```
contract("Election",function(accounts){it("initialize with two candidate",function(){return Election.deployed().  
then(function(instance){return instance.candidateCount();}).then(function(count){assert.equal(count,2);}});});
```


Prog. 3 MetaMask Configuration Steps and Transaction

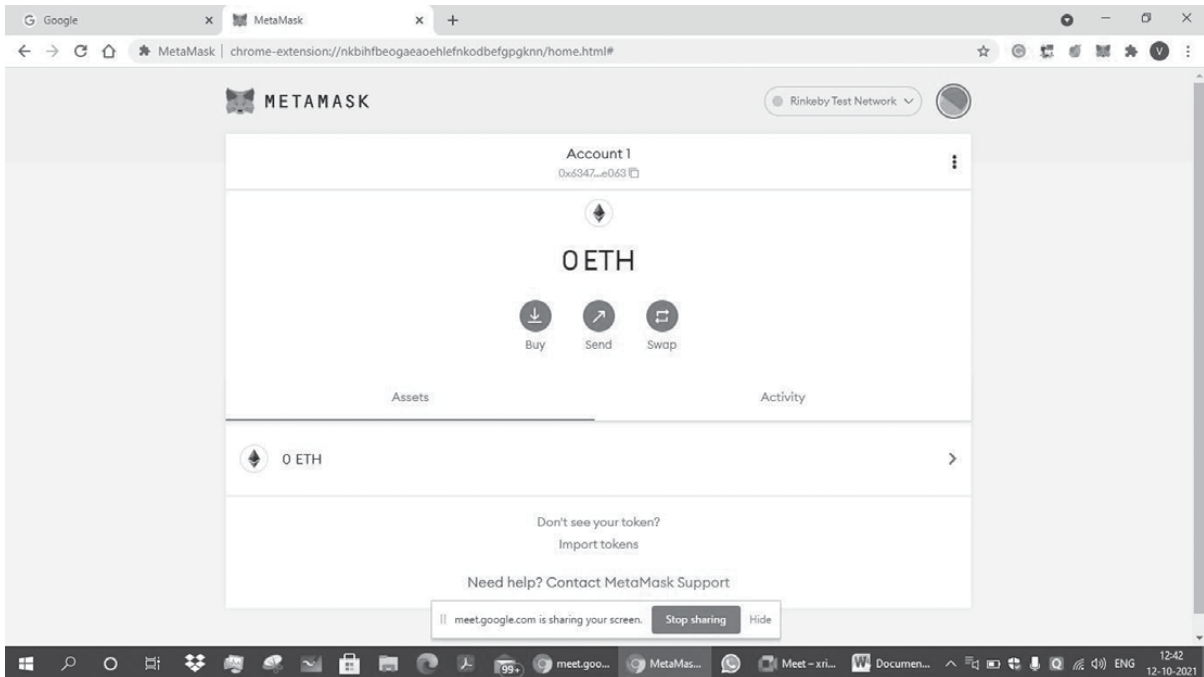


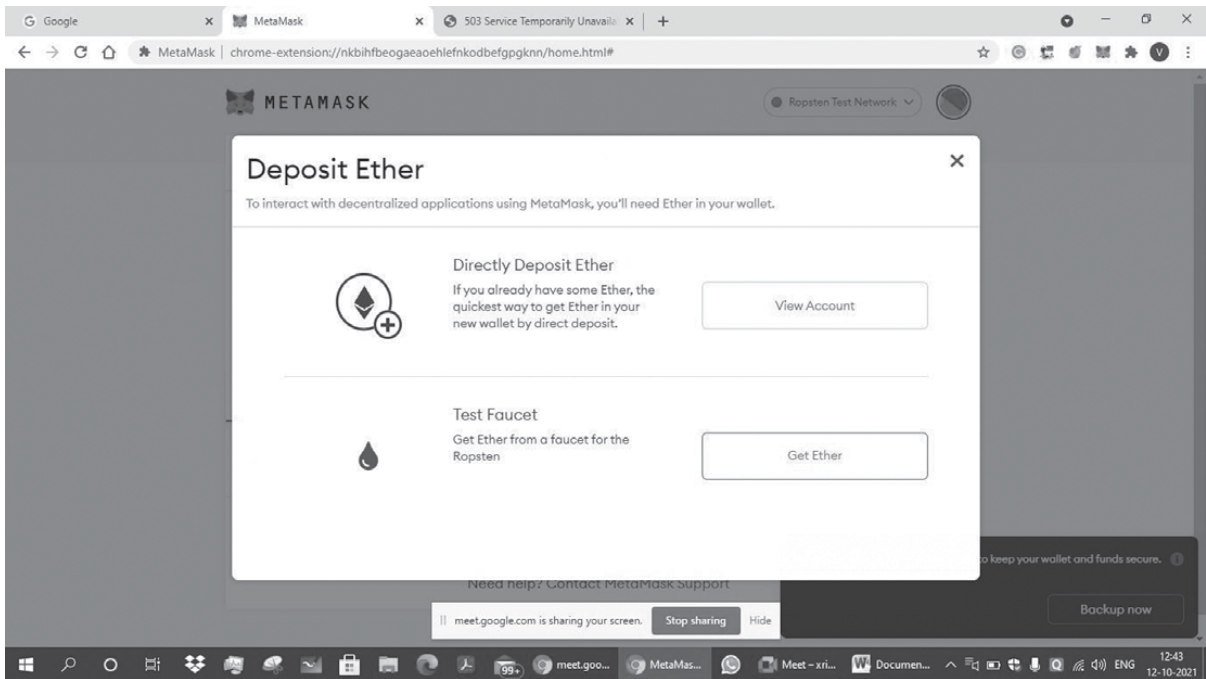
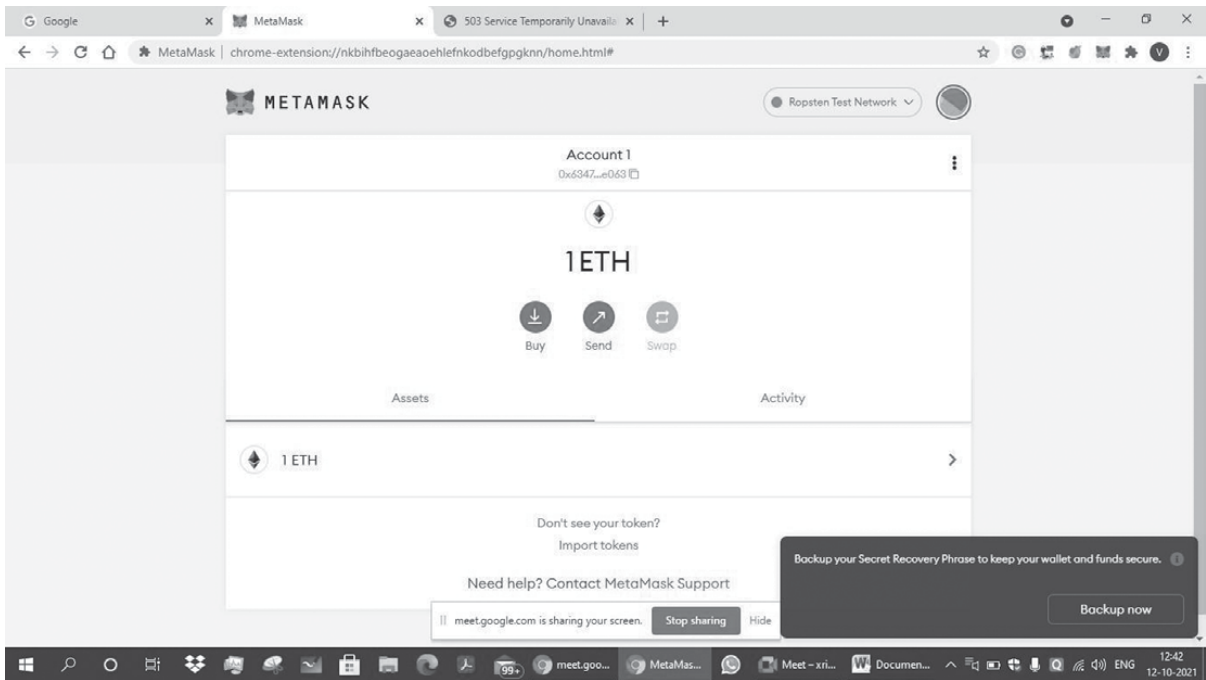






b84ced9f4968a896a38de6828be39f2811b9ce6109253b8509d17ee33fa4ff44



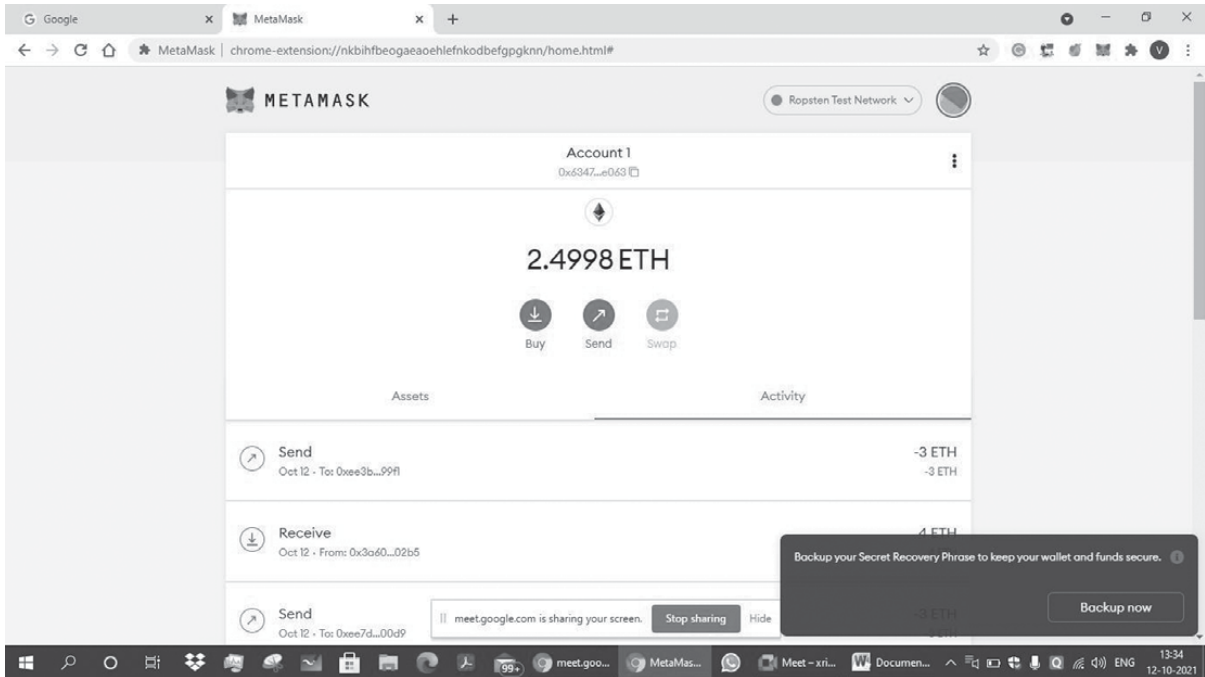
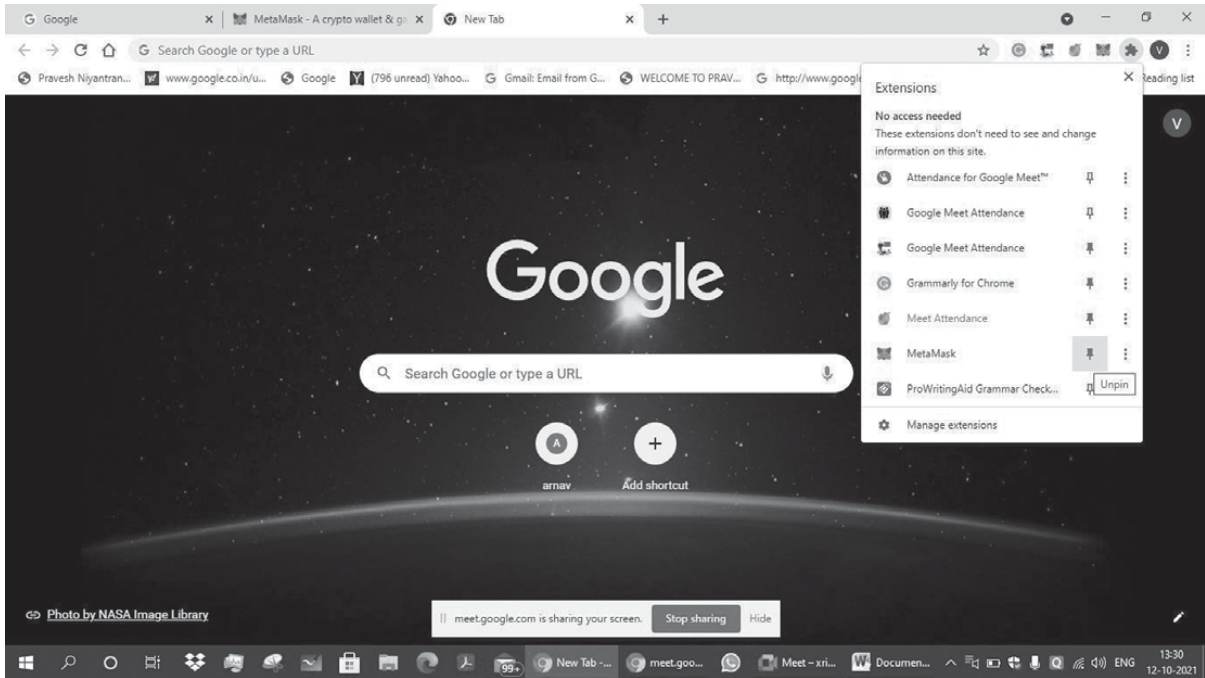


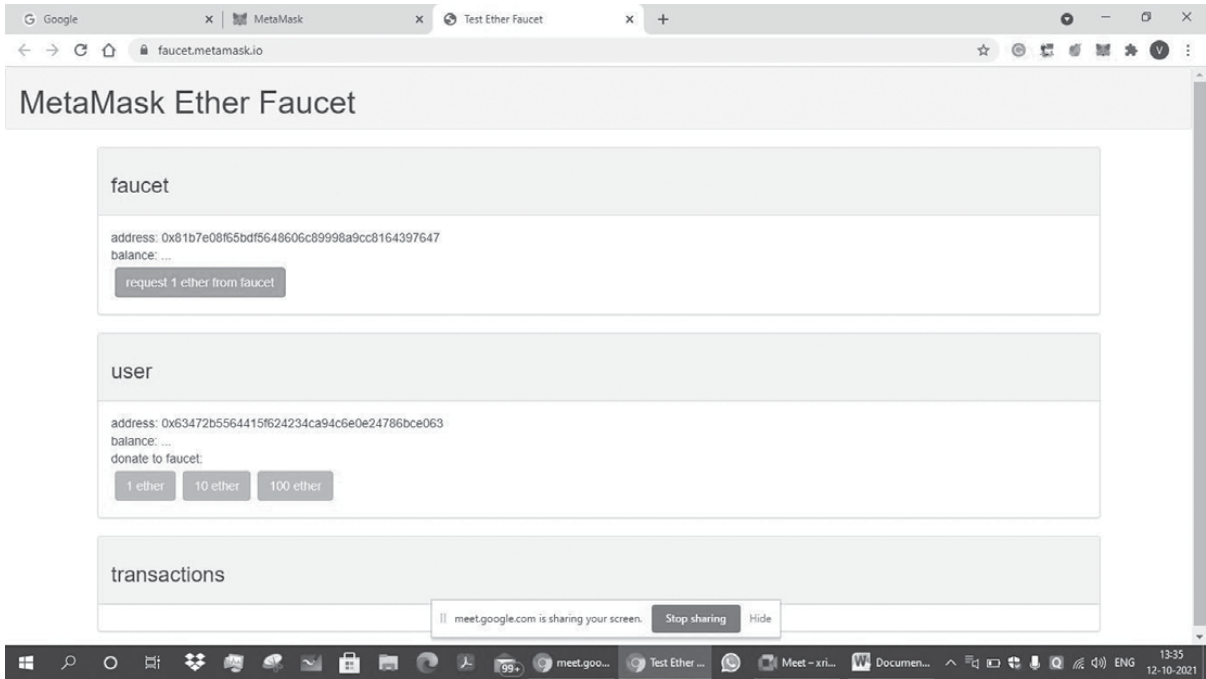
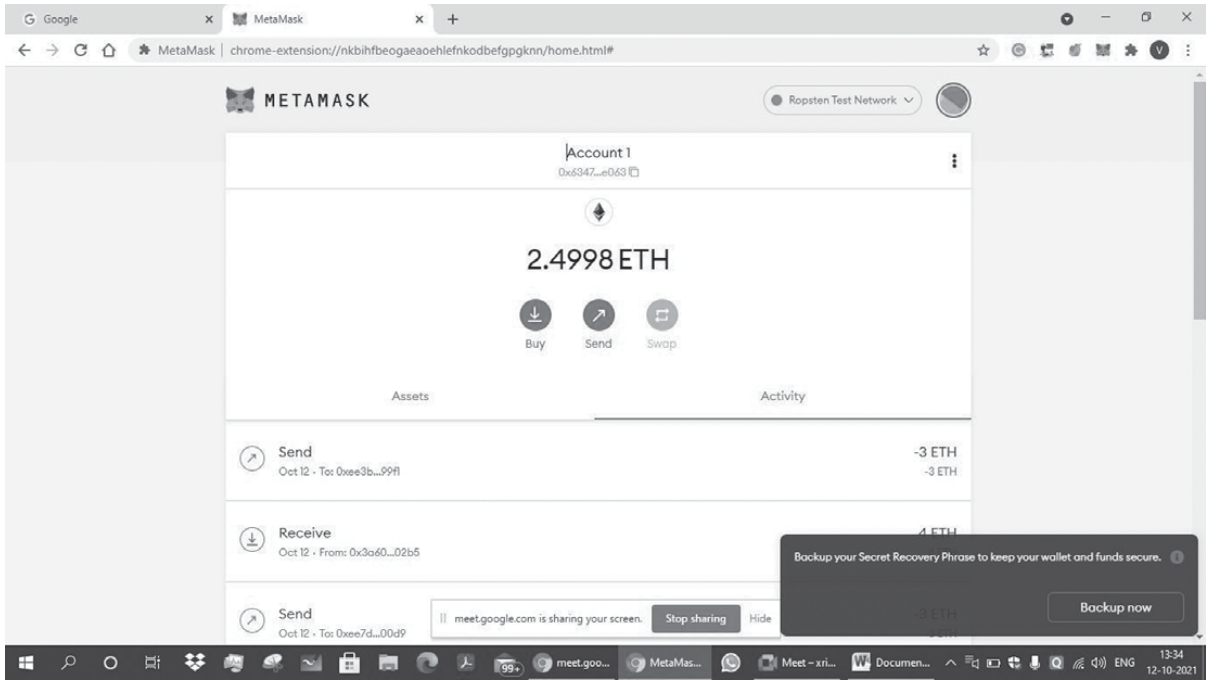
0x63472B5564415F624234cA94c6E0E24786bce063

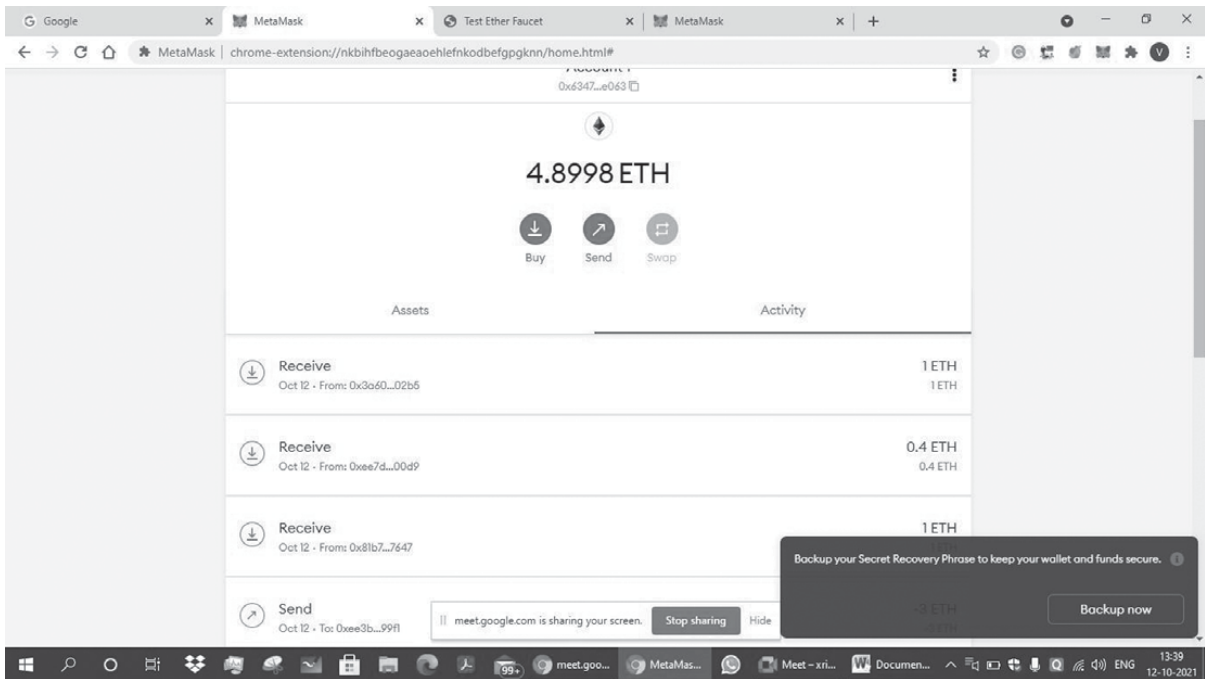
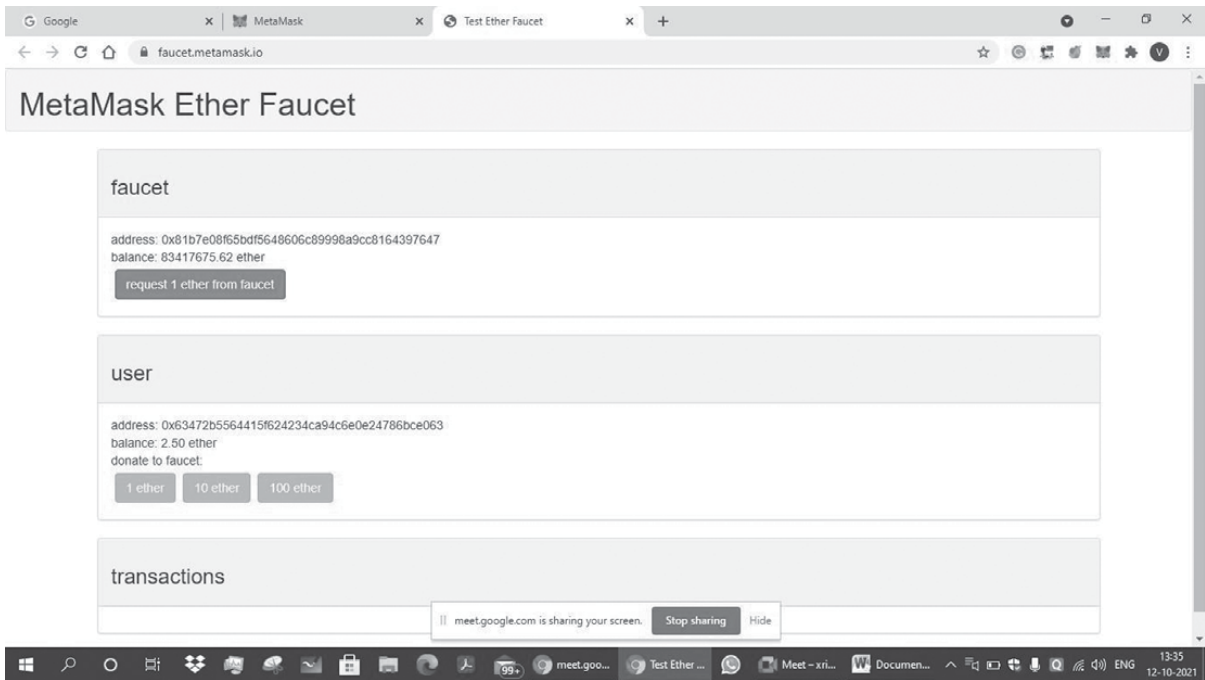
0x63472B5564415F624234cA94c6E0E24786bce063

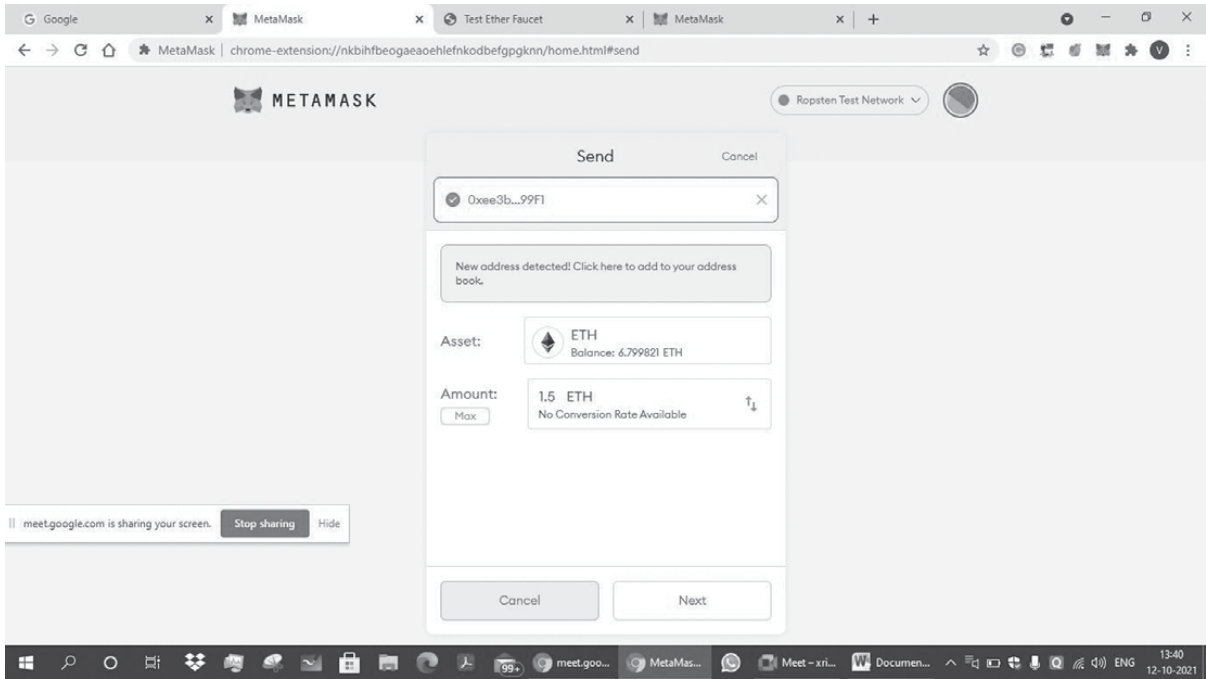
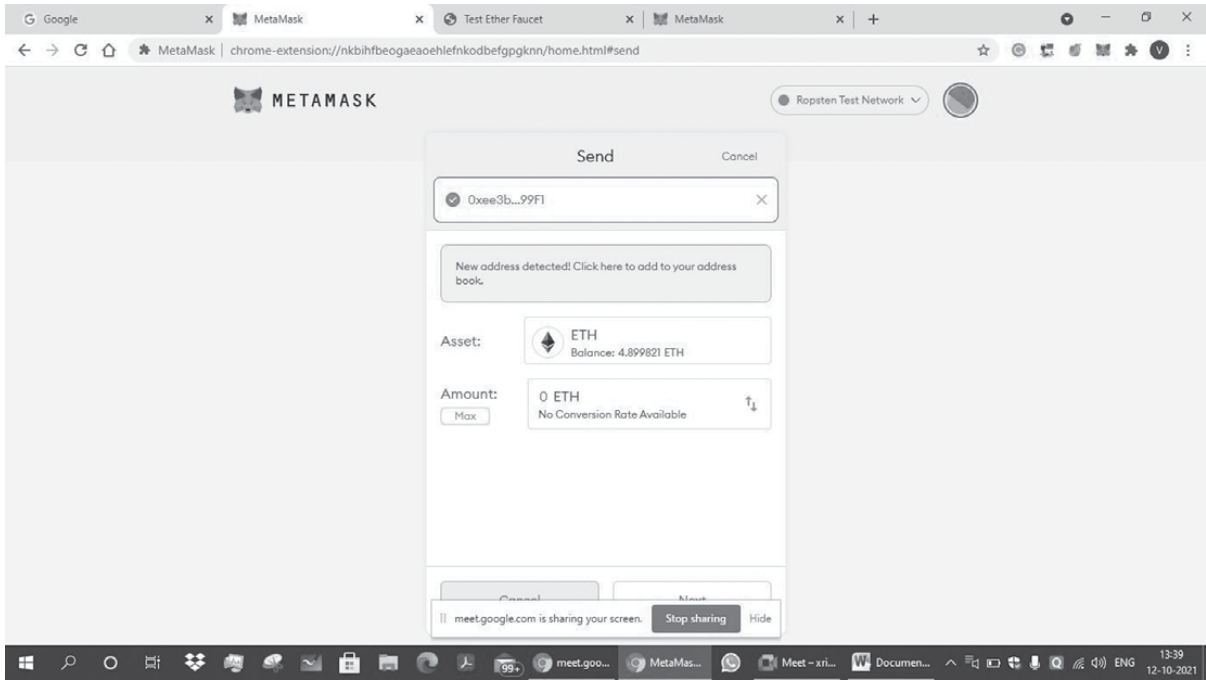
0x63472b5564415f624234ca94c6e0e24786bce063

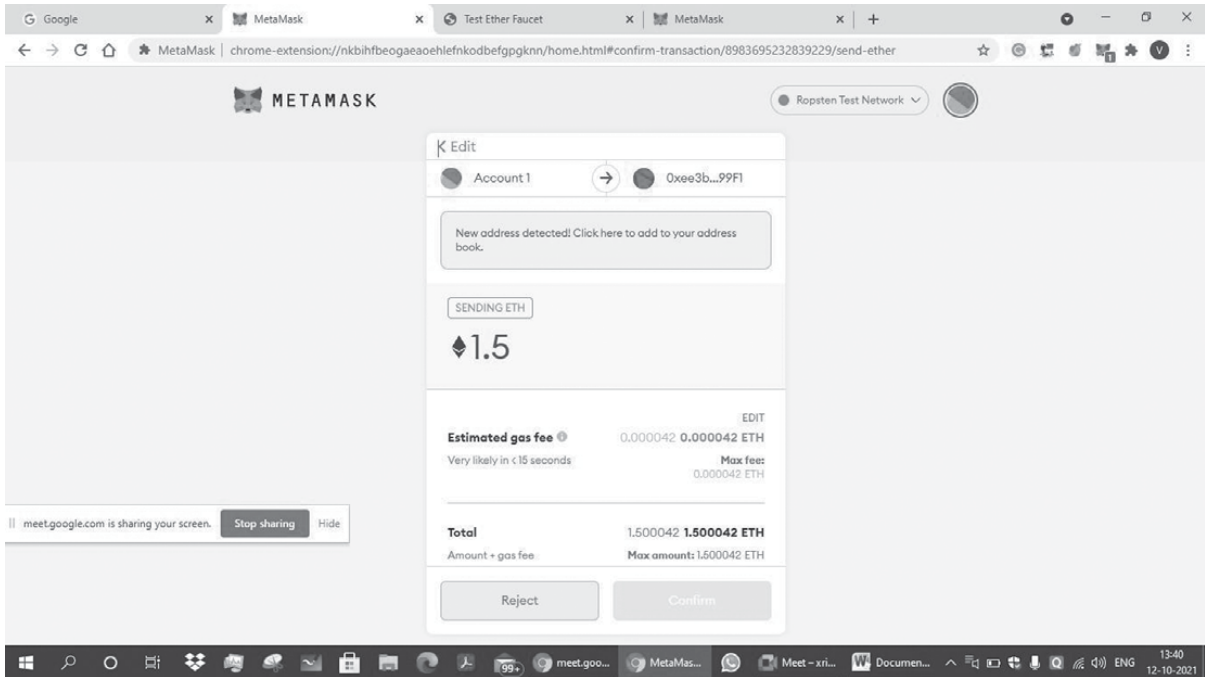
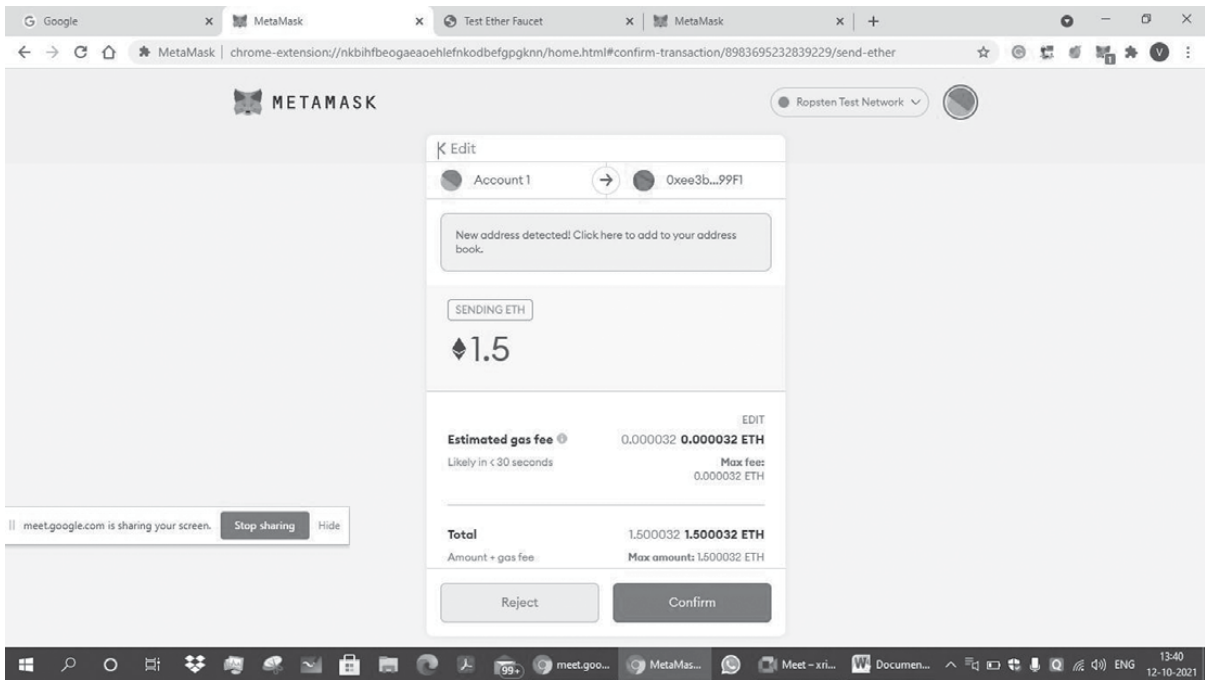
b84ced9f4968a896a38de6828be39f2811b9ce6109253b8509d17ee33fa4ff44

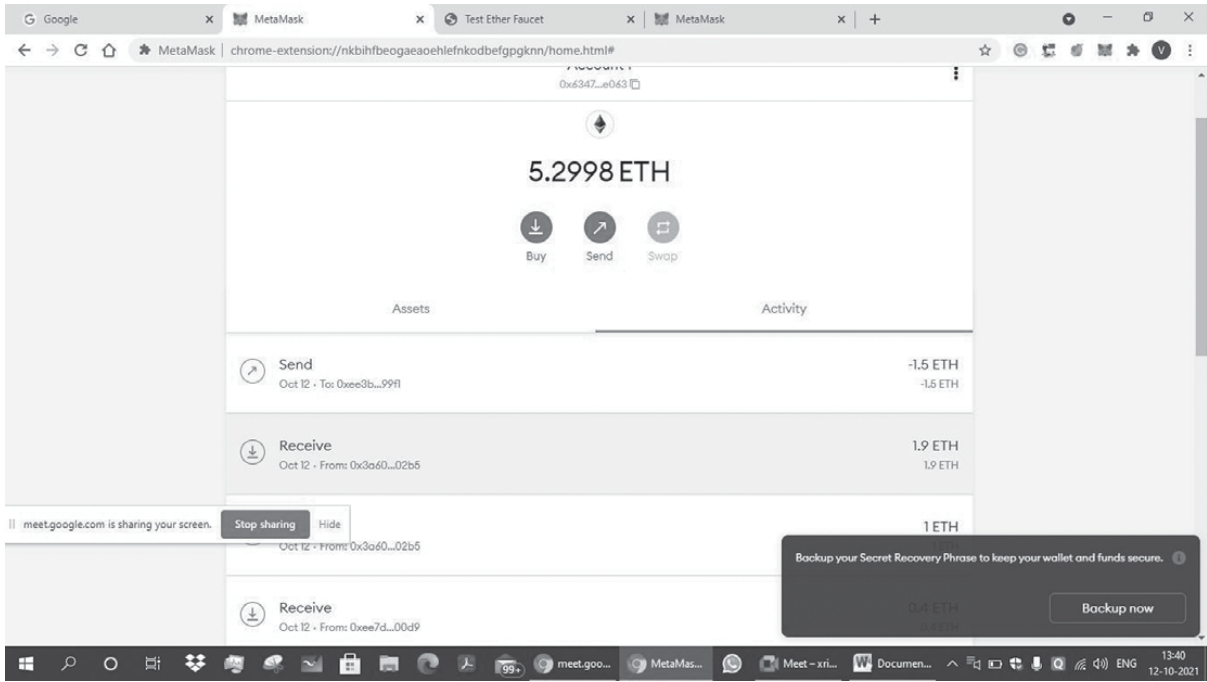












Prog. 4 Blockchain Cryptography Program using Java

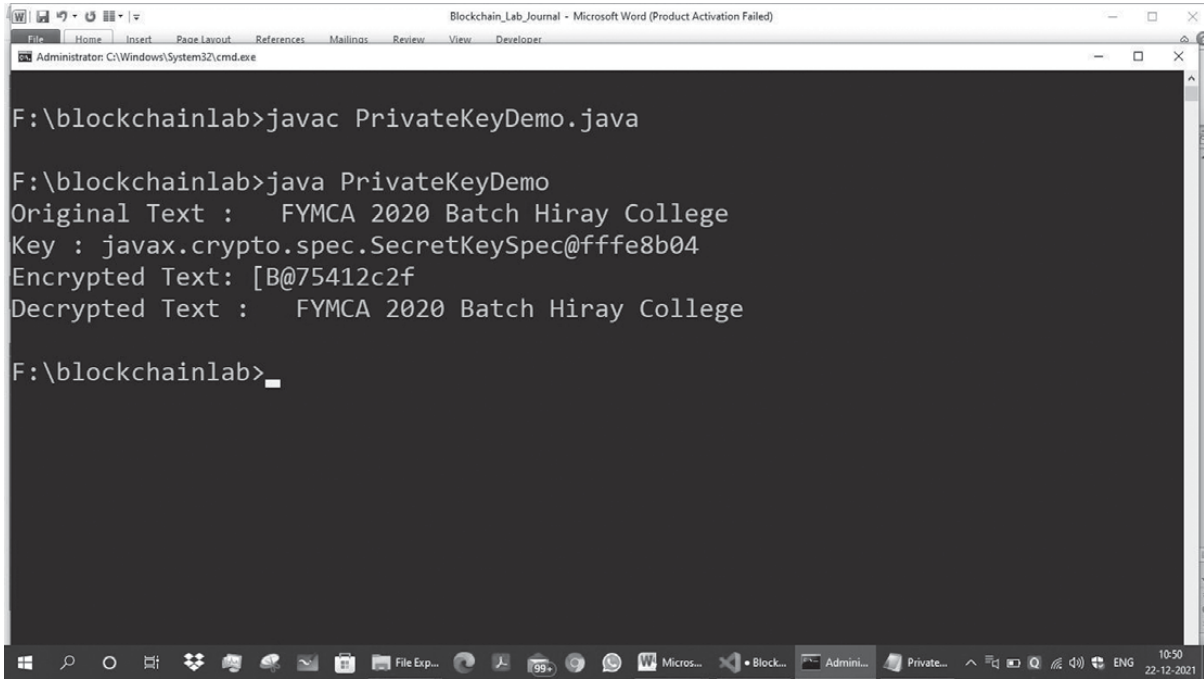
```
import java.security.*; import javax.crypto.*;
/* Use of Private Key in Encryption*/

public class PrivateKeyDemo
{
    static Key key;
    static Cipher cipher;
    static String algorithm = "AES";
    public static void main(String args[]) throws Exception
    {
        key = KeyGenerator.getInstance(algorithm).generateKey();
        cipher = Cipher.getInstance(algorithm);
        String text = "FYMCA 2020 Batch Hiray College";
        byte[] encryptionBytes = encrypt(text);
        System.out.println("Original Text: "+text);
        System.out.println("Key: "+key.toString());
        System.out.println("Encrypted Text: "+encrypt(text));
        System.out.println("Decrypted Text: "+decrypt(encryptionBytes));
    } // End of Main Method

    public static byte[] encrypt(String input) throws Exception
    {
        cipher.init(Cipher.ENCRYPT_MODE, key);
        byte [] inputBytes = input.getBytes();
        return cipher.doFinal(inputBytes);
    } // close of encrypt method

    public static String decrypt(byte[] encryptionBytes) throws Exception
    {
        cipher.init(Cipher.DECRYPT_MODE, key);
        byte[] convertDecrypt = cipher.doFinal(encryptionBytes);
        String convertString = new String(convertDecrypt);
        return convertString;
    } // close decrypt method
}
```

Output:



```
Administrator: C:\Windows\System32\cmd.exe

F:\blockchainlab>javac PrivateKeyDemo.java

F:\blockchainlab>java PrivateKeyDemo
Original Text :   FYMCA 2020 Batch Hiray College
Key : javax.crypto.spec.SecretKeySpec@fffe8b04
Encrypted Text: [B@75412c2f
Decrypted Text :   FYMCA 2020 Batch Hiray College

F:\blockchainlab>_
```

Prog. 5 Blockchain Public Key Cryptography

```
import java.security.*;
import javax.crypto.*;
import java.util.*;
// Public Key Example
public class PublicKeyDemo
{
    static KeyPair keyPair;
    static String algorithm = "RSA";
    public static void main(String args[]) throws Exception
    {
        KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance(algorithm);

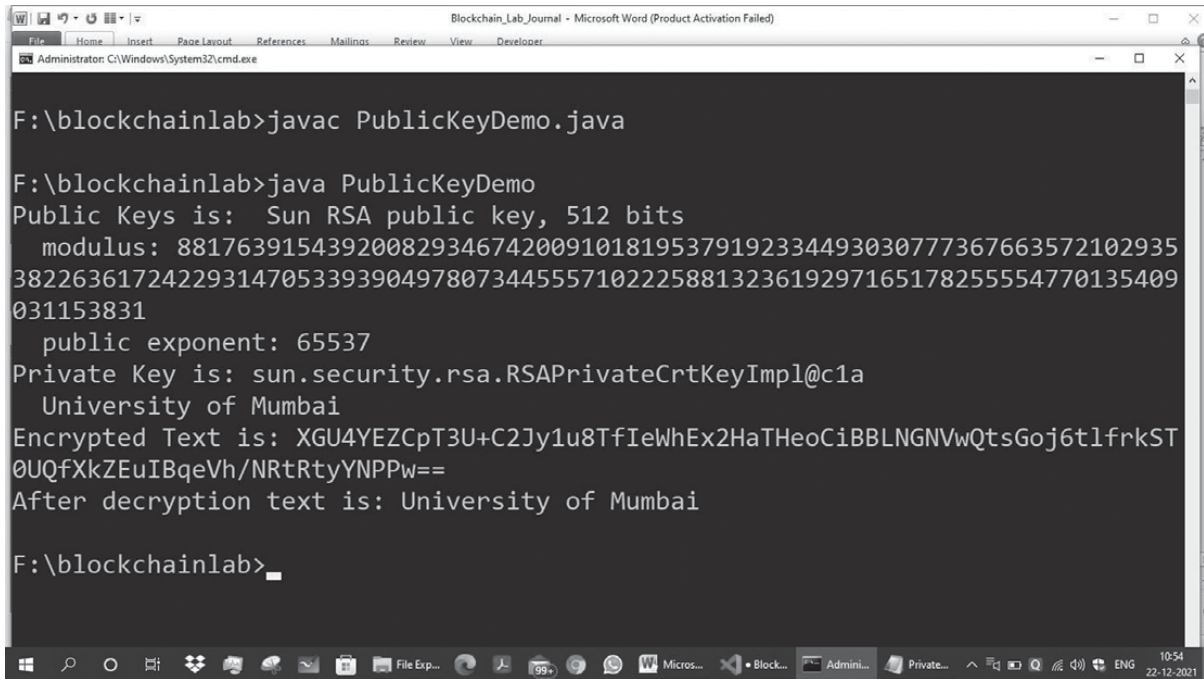
        keyPairGenerator.initialize(512);
        keyPair = keyPairGenerator.generateKeyPair();

        Cipher cipher = Cipher.getInstance(algorithm);
        String text = " University of Mumbai";

        System.out.println("Public Keys is: "+keyPair.getPublic().toString());
        System.out.println("Private Key is: "+keyPair.getPrivate().toString());
        System.out.println(" "+text);

        // Encrypt the text
        cipher.init(Cipher.ENCRYPT_MODE, keyPair.getPublic());
        byte[] encryptedText = cipher.doFinal(text.getBytes());
        String ciphertext = new String(Base64.getEncoder().encode(encryptedText));
        System.out.println("Encrypted Text is: "+ciphertext);
        // Decryption using Private key
        cipher.init(Cipher.DECRYPT_MODE, keyPair.getPrivate());
        byte[] ciphertextByte = Base64.getDecoder().decode(ciphertext.getBytes()); byte[] decryptedByte = cipher.
doFinal(ciphertextByte);
        String decryptedString = new String(decryptedByte);
        System.out.println("After decryption text is:"+decryptedString);
    }
}
```

Output:



```
F:\blockchainlab>javac PublicKeyDemo.java

F:\blockchainlab>java PublicKeyDemo
Public Keys is: Sun RSA public key, 512 bits
  modulus: 8817639154392008293467420091018195379192334493030777367663572102935
382263617242293147053393904978073445557102225881323619297165178255554770135409
031153831
  public exponent: 65537
Private Key is: sun.security.rsa.RSAPrivateCrtKeyImpl@c1a
  University of Mumbai
Encrypted Text is: XGU4YEZCpT3U+C2Jy1u8TfIeWhEx2HaTHeoCiBBLNGNVwQtsGoj6tlfrkST
0UQfXkZEuIBqeVh/NRtRtyYNPPw==
After decryption text is: University of Mumbai

F:\blockchainlab>
```