



**R.D.&S.H NATIONAL COLLEGE & S.W.A. SCIENCE
COLLEGE**
Bandra, Mumbai - 400050

DEPARTMENT OF COMPUTER SCIENCE

M.Sc. Computer Science - Semester III

**Web3 Technologies Practical Journal
2024-2025**

Seat No. 1293026



R.D. & S.H. NATIONAL COLLEGE & S. W.A. SCIENCE COLLEGE, Bandra, Mumbai - 400050.



Department of Computer Science

CERTIFICATE

This is to certify that Mr/Ms. Jishan Farid Ahmed Shaikh of M.Sc Part II (Sem III) class has satisfactorily completed VIII Practicals in the subject of Web3 Technologies Practical as a part of M.Sc. Degree Course in Computer Science during the academic year 2024 – 2025.

Date of Submission:

Faculty Incharge

Co-ordinator,
Department of Computer Science

Signature of External Examiner

Practical

Web3 Technologies Practical

Index

Sr.	Title	Page	Date	Sign
1	Install and understand Docker container, Node.js, Java and Hyperledger Fabric, Ethereum and perform necessary software installation on local machine/create instance on Cloud to run.	2		
2	Create and deploy a block chain network using Hyperledger Fabric SDK for Java	7		
3	Interact with a block chain network. Execute transactions and requests against a block chain network by creating an app to test the network and its rules	10		
4	Deploy an asset-transfer app using block chain. Learn app development within a Hyperledger Fabric network.	17		
5	Use block chain to track fitness club rewards.	22		
6	Build a web app that uses Hyperledger Fabric to track and trace member rewards.	27		
7	Car auction network: A Hello World example with Hyperledger Fabric Node SDK and IBM Block chain Starter Plan. Use Hyperledger Fabric to invoke chaincode while storing results and data in the starter plan	31		
8	Develop a voting application using Hyperledger and Ethereum.	37		

Practical 1

Aim: Install and understand Docker container, Node.js, Java and Hyperledger Fabric, Ethereum and perform necessary software installation on local machine/create instance on Cloud to run.

Docker Container:

- **Definition:** Docker is a platform used to develop, ship, and run applications inside lightweight containers.
- **Purpose:** It ensures consistency across multiple environments (development, testing, production).
- **Usage:** Containers package software and its dependencies, so it can run anywhere.
- **Key Concept:** Docker images are templates for containers, and Docker containers are instances of these images.

Node.js:

- **Definition:** Node.js is an open-source, cross-platform runtime environment that executes JavaScript code server-side.
- **Purpose:** It is designed for building scalable network applications.
- **Usage:** It leverages the non-blocking, event-driven architecture to handle concurrent connections efficiently.
- **Key Concept:** Node.js is widely used for web servers, REST APIs, and real-time applications like chat apps.

Java:

- **Definition:** Java is a high-level, object-oriented programming language used for building platform-independent applications.
- **Purpose:** Java allows developers to write code once and run it anywhere (cross-platform).
- **Usage:** It's used in various domains, including web development, mobile apps (Android), enterprise systems, and more.
- **Key Concept:** Java programs are compiled to bytecode, which runs on the Java Virtual Machine (JVM).

Hyperledger Fabric:

- **Definition:** Hyperledger Fabric is a permissioned blockchain framework designed for enterprise use.
- **Purpose:** It provides a modular architecture for building blockchain solutions.
- **Usage:** It allows businesses to create private, secure, and scalable blockchain networks.
- **Key Concept:** Hyperledger Fabric supports smart contracts (chaincode), private transactions, and decentralized consensus mechanisms.

Ethereum:

- **Definition:** Ethereum is a decentralized, open-source blockchain platform that supports smart contracts and decentralized applications (DApps).
- **Purpose:** Ethereum provides a blockchain environment for developing and deploying smart contracts.
- **Usage:** It's widely used for building decentralized finance (DeFi) applications and token systems like NFTs.

- **Key Concept:** Ethereum runs on its native cryptocurrency, Ether (ETH), and uses a proof-of-work (PoW) consensus mechanism, with a transition to proof-of-stake (PoS).

Software Installation and Setup:

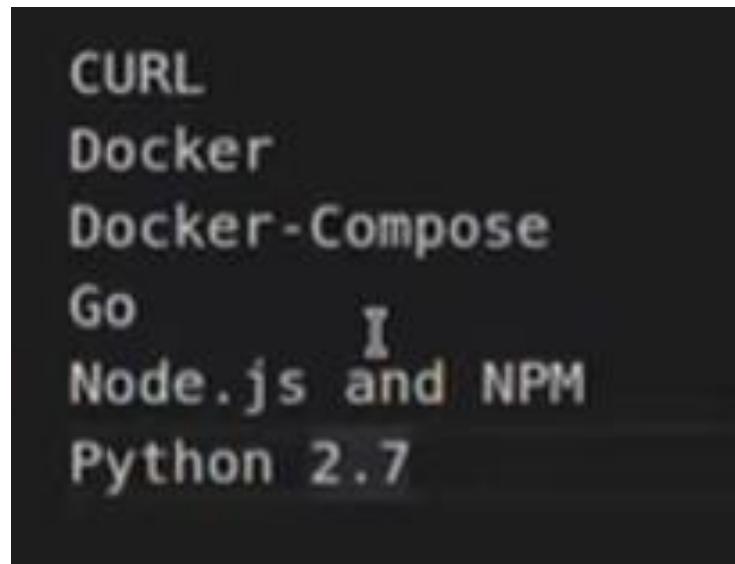
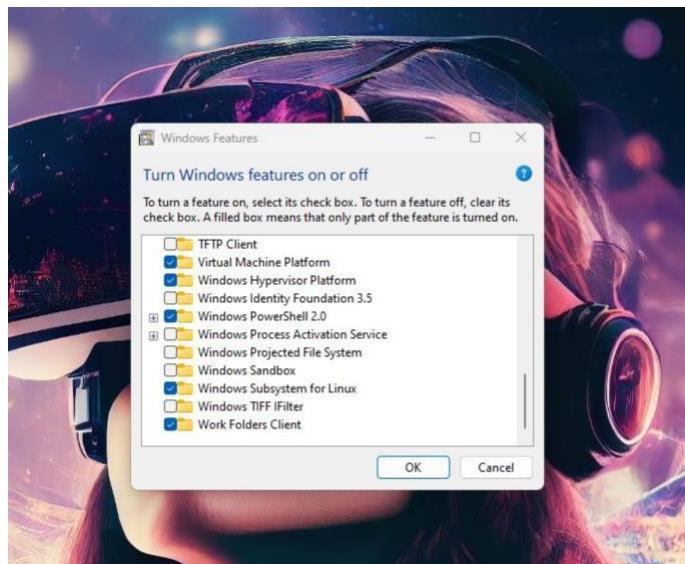
- **Local Machine Setup:** Install Docker, Node.js, Java, and any required dependencies for running the above technologies.
- **Cloud Instance Setup:** Set up a cloud instance (e.g., AWS, Azure) to run Docker containers or host blockchain networks like Hyperledger Fabric and Ethereum

Turn on windows Feature

Windows subsystem for linux

Window Virtual Machine

Windows Hypervisor



Sudo apt install curl

Download Docker desktop

Sudo apt install docker

Sudo apt install docker-compose

Sudo apt install golang-go

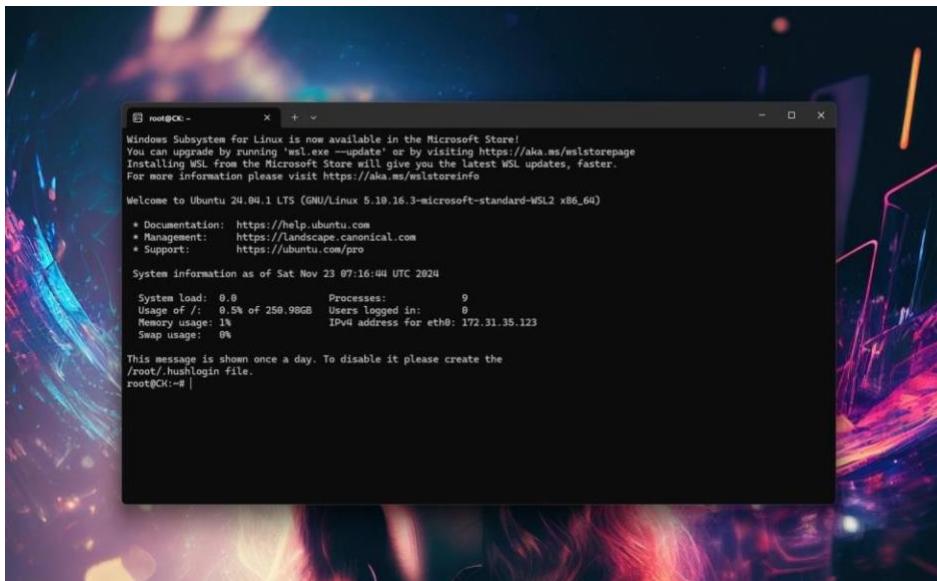
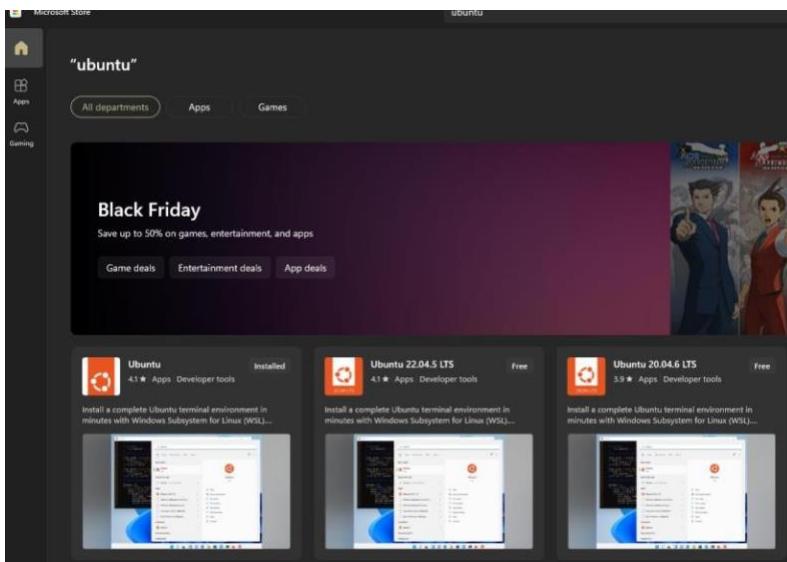
Sudo apt install pyhton

Sudo apt install nodejs

sudo apt install npm

To check version name –version

Install Ubuntu from store



```
/root
root@CK:~# sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.4).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
root@CK:~# curl --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3 brotli/1.1.0 zstd/1.5.5 libidn2/2.3.7 libpsl/0.21
.2 (+libidn2/2.3.7) libssh/0.10.6/openssl/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/2.6.7
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.4
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp sftp smb smb
s smtp smtps telnet tftp
Features: alt-svc AsyncDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM PSL SPNEGO SSL t
hreadsafe TLS-SRP UnixSockets zstd
root@CK:~# |
```

```
root@CK:~# python3 --version
Python 3.12.3
root@CK:~# |
```

```
root@CK:~# docker --version
Docker version 27.3.1, build ce12230
root@CK:~# docker-compose --version
Docker Compose version v2.29.7-desktop.1
root@CK:~#
```

```
root@CK:~# sudo apt install golang-go
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc
  gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-14-base gcc-x86-64-linux-gnu golang-1.22-go golang-1.22-src
  golang-src libaom3 libasan8 libatomic1c1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libde265-0
  libgcc-13-dev libgcc-s1 libgd3 libgomp1 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1
  libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libpkgconf3 libquadmath0 libstdc++-13-dev libstdc++6 libtsan2 libubsan1
  libxpm4 linux-libc-dev manpages-dev pkg-config pkgconf pkgconf-bin rpcsvc-proto
Suggested packages:
  cpp-doc gcc-13-locales cpp-13-doc g++-multilib g++-13-multilib gcc-13-doc gcc-multilib make autoconf automake
  libtool flex bison gdb gcc-doc gcc-13-multilib gdb-x86-64-linux-gnu bzr | brz mercurial subversion glibc-doc
  libgd-tools libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc
  libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e libheif-plugin-svtenc libstdc++-13-doc
The following NEW packages will be installed:
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc
  gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu golang-1.22-go golang-1.22-src golang-go golang-src
  libaom3 libasan8 libatomic1c1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libde265-0 libgcc-13-dev
  libgd3 libgomp1 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libhwasan0 libisl23
  libitm1 liblsan0 libmpc3 libpkgconf3 libquadmath0 libstdc++-13-dev libtsan2 libubsan1 libxpm4 linux-libc-dev
  manpages-dev pkg-config pkgconf pkgconf-bin rpcsvc-proto
The following packages will be upgraded:
  gcc-14-base libgcc-s1 libstdc++6
3 upgraded, 50 newly installed, 0 to remove and 58 not upgraded.
```

```
apt install nodejs
root@CK:~# apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcares2 libnode109 libuv1.64 node-acorn node-busboy node-cjs-module-lexer node-undici node-xtend nodejs-doc
Suggested packages:
  npm
The following NEW packages will be installed:
  libcares2 libnode109 libuv1.64 node-acorn node-busboy node-cjs-module-lexer node-undici node-xtend nodejs-doc
0 upgraded, 10 newly installed, 0 to remove and 58 not upgraded.
Need to get 16.2 MB of archives.
After this operation, 70.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] |
```

Node -v

sudo apt install npm

```
root@CK:~# sudo apt install npm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
npm is already the newest version (9.2.0~ds1-2).
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
root@CK:~# |
```

Install Hyperledger fabric

```
git config --global core.autocrlf false
```

```
git config --global core.longpaths true
```

```
root@CK:~# git config --global core.autocrlf false
error: unknown non-ascii option in string: '--global'
usage: git config [<options>]

Config file location
--[no-]global      use global config file
--[no-]system      use system config file
--[no-]local       use repository config file
```

```
root@CK:~# git config --global core.longpaths true
root@CK:~# |
```

```
Mkdir rkprac1
```

```
cd rkprac1
```

```
curl -sSL http://bit.ly/2ysbOFE | bash -s
```

```
root@CK:~/rkprac1      x + v
root@CK:~# mkdir rkprac1
root@CK:~# cd rkprac1
root@CK:~/rkprac1# $ curl false sSL http://bit.ly/2ysbOFE | bash -s
$: command not found
root@CK:~/rkprac1# curl false sSL http://bit.ly/2ysbOFE | bash -s
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left Speed
0       0       0       0       0       0  --:--:--  0:00:01  --:--:--     0|
```

```
root@CK:~/rkprac1      x + v
root@CK:~/rkprac1# curl -sSL http://bit.ly/2ysbOFE | bash -s
Clone hyperledger/fabric-samples repo
====> Cloning hyperledger/fabric-samples repo
Cloning into 'fabric-samples'...
remote: Enumerating objects: 14091, done.
remote: Counting objects: 100% (123/123), done.
remote: Compressing objects: 100% (82/82), done.
Receiving objects: 67% (9561/14091), 10.50 MiB | 10.49 MiB/s
```

Practical 2

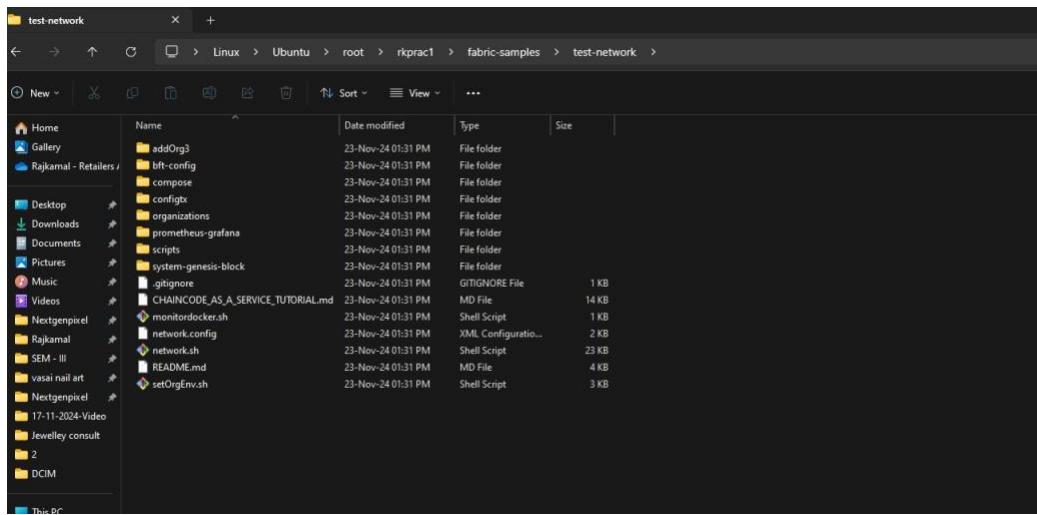
Aim: Create and deploy a block chain network using Hyperledger Fabric SDK for Java

BUILD THE FIRST HYPERLEDGER NETWORK

Locate network.sh file

to get all component

cd fabric-samples/test-network/



Cat network.sh

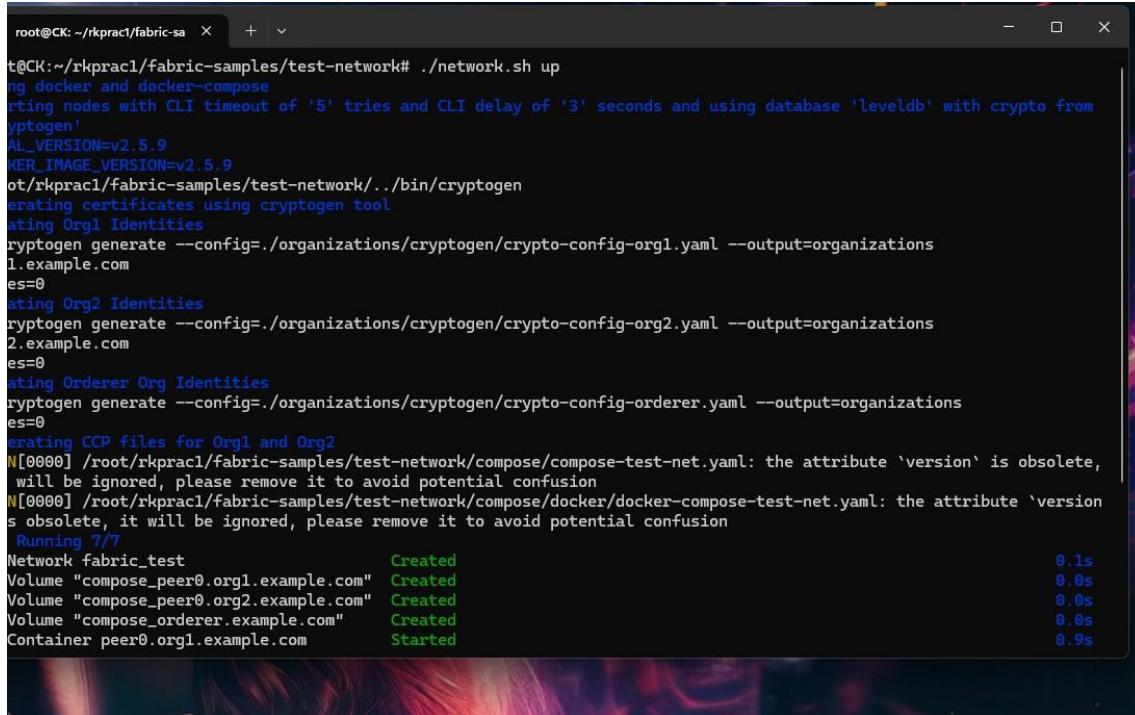
```
root@CK:~/rkprac1/fabric-samples/test-network# cat network.sh
#!/bin/bash
#
# Copyright IBM Corp All Rights Reserved
#
# SPDX-License-Identifier: Apache-2.0
#
# This script brings up a Hyperledger Fabric network for testing smart contracts
# and applications. The test network consists of two organizations with one
# peer each, and a single node Raft ordering service. Users can also use this
# script to create a channel deploy a chaincode on the channel
#
# prepending $PWD/../bin to PATH to ensure we are picking up the correct binaries
# this may be commented out to resolve installed version of tools if desired
#
# However using PWD in the path has the side effect that location that
# this script is run from is critical. To ease this, get the directory
# this script is actually in and infer location from there. (putting first)
ROOTDIR=$(cd "$(dirname "$0")" && pwd)
export PATH=$ROOTDIR/.:/bin:$PWD/..:bin:$PATH
export FABRIC_CFG_PATH=$PWD/configtx
export VERBOSE=false

# push to the required directory & set a trap to go back if needed
pushd ${ROOTDIR} > /dev/null
trap "popd > /dev/null" EXIT

. scripts/utils.sh
```

To Up your network

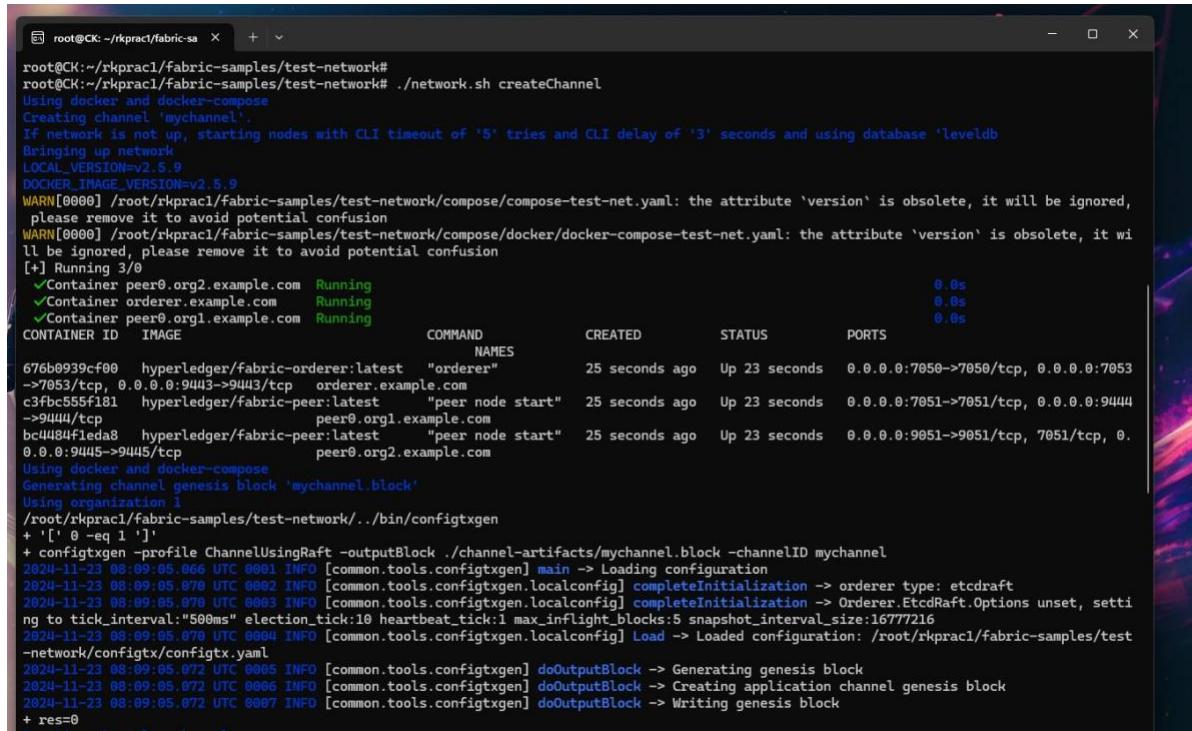
./network.sh up



```
t@CK:~/rkprac1/fabric-samples/test-network# ./network.sh up
ng docker and docker-compose
rtting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from
yptogen'
AL_VERSION=v2.5.9
CKER_IMAGE_VERSION=v2.5.9
ot/rkprac1/fabric-samples/test-network/../bin/cryptogen
erating certificates using cryptogen tool
ating Org1 Identities
ryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
1.example.com
es=0
ating Org2 Identities
ryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
2.example.com
es=0
ating Orderer Org Identities
ryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
es=0
erating CCP files for Org1 and Org2
N[0000] /root/rkprac1/fabric-samples/test-network/compose/compose-test-net.yaml: the attribute 'version' is obsolete,
will be ignored, please remove it to avoid potential confusion
N[0000] /root/rkprac1/fabric-samples/test-network/compose/docker/docker-compose-test-net.yaml: the attribute 'version
s obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 7/7
Network fabric_test
Volume "compose_peer0.org1.example.com" Created
Volume "compose_peer0.org2.example.com" Created
Volume "compose_orderer.example.com" Created
Container peer0.org1.example.com Started
          Created          0.1s
          Created          0.0s
          Created          0.0s
          Started          0.9s
```

Creating a channel- create channels between two nodes of the network

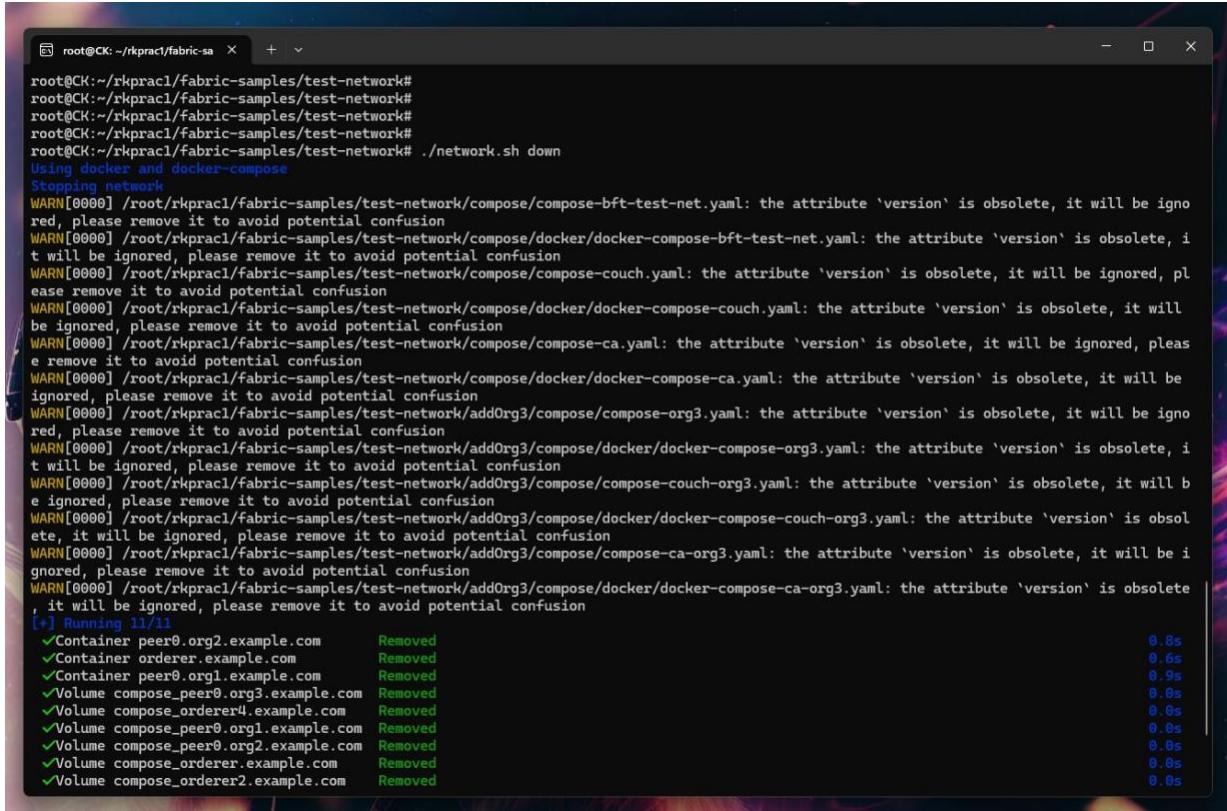
./network.sh createChannel



```
root@CK:~/rkprac1/fabric-samples/test-network#
root@CK:~/rkprac1/fabric-samples/test-network# ./network.sh createChannel
Using docker and docker-compose
Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb
Bringing up network
LOCAL_VERSION=v2.5.9
DOCKER_IMAGE_VERSION=v2.5.9
WARN[0000] /root/rkprac1/fabric-samples/test-network/compose/compose-test-net.yaml: the attribute 'version' is obsolete, it will be ignored,
please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/fabric-samples/test-network/compose/docker/docker-compose-test-net.yaml: the attribute 'version' is obsolete, it wi
ll be ignored, please remove it to avoid potential confusion
[+] Running 3/0
✓ Container peer0.org2.example.com Running
✓ Container orderer.example.com Running
✓ Container peer0.org1.example.com Running
CONTAINER ID IMAGE COMMAND NAMES CREATED STATUS PORTS
676b0939cf00 hyperledger/fabric-orderer:latest "orderer" 25 seconds ago Up 23 seconds 0.0.0.0:7050->7050/tcp, 0.0.0.0:7053
->7053/tcp, 0.0.0.0:9443->9443/tcp orderer.example.com
c3fbcb55f181 hyperledger/fabric-peer:latest "peer node start" 25 seconds ago Up 23 seconds 0.0.0.0:7051->7051/tcp, 0.0.0.0:9444
->9444/tcp peer0.org1.example.com
bc4484fleda8 hyperledger/fabric-peer:latest "peer node start" 25 seconds ago Up 23 seconds 0.0.0.0:9051->9051/tcp, 7051/tcp, 0.
0.0.0:9445->9445/tcp peer0.org2.example.com
Using docker and docker-compose
Generating channel genesis block 'mychannel.block'
Using organization 1
/root/rkprac1/fabric-samples/test-network/../bin/configtxgen
+ '[' 0 -eq 1 ']'
+ configtxgen -profile ChannelUsingRaft -outputBlock ./channel-artifacts/mychannel.block -channelID mychannel
2024-11-23 08:09:05.066 UTC 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2024-11-23 08:09:05.070 UTC 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdraft
2024-11-23 08:09:05.079 UTC 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Options unset, setti
ng to tick_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2024-11-23 08:09:05.076 UTC 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /root/rkprac1/fabric-samples/test
-network/configtx/configtx.yaml
2024-11-23 08:09:05.075 UTC 0005 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2024-11-23 08:09:05.072 UTC 0006 INFO [common.tools.configtxgen] doOutputBlock -> Creating application channel genesis block
2024-11-23 08:09:05.072 UTC 0007 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
+ res=0
```

To turn down your network

./network.sh down



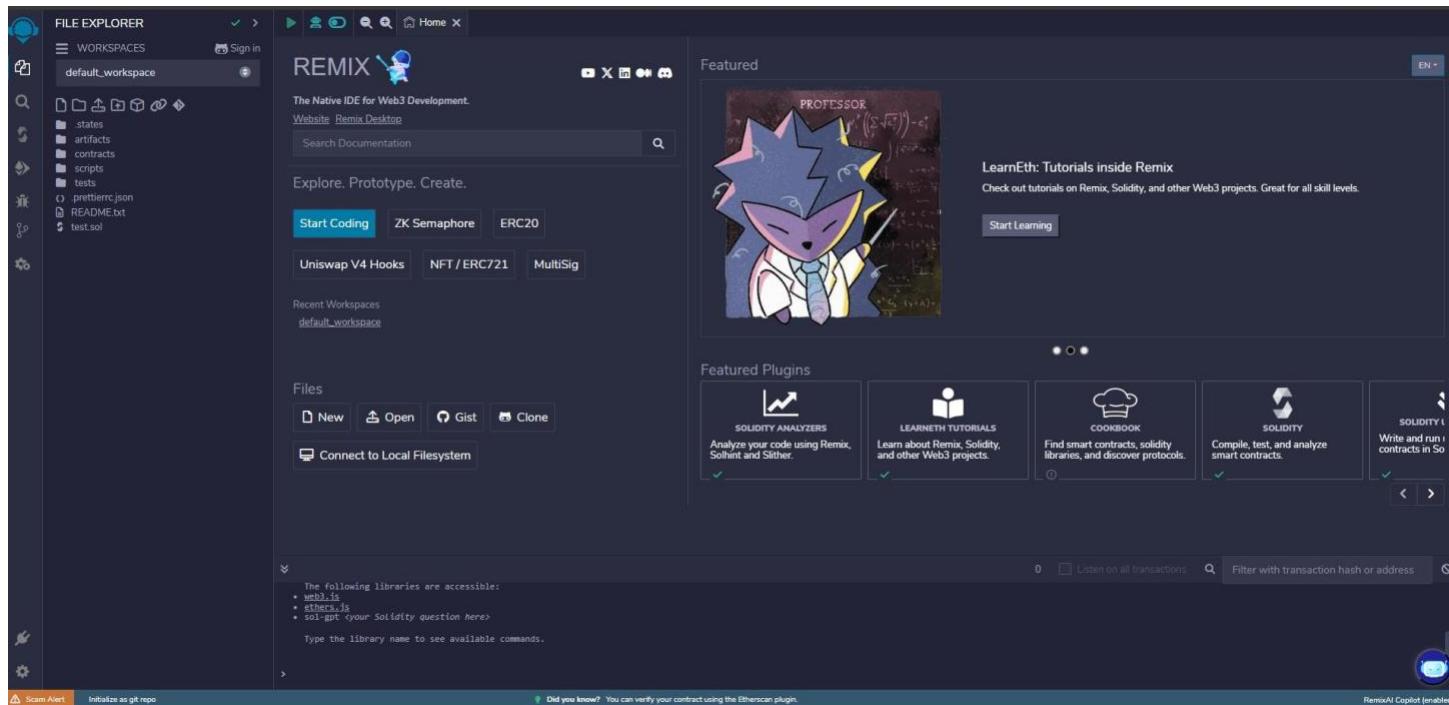
The screenshot shows a terminal window with the following output:

```
root@CK:~/rkprac1/fabric-samples/test-network# ./network.sh down
Using docker and docker-compose
Stopping network
WARN[0000] /root/rkprac1/fabric-samples/test-network/compose/compose-bft-test-net.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/fabric-samples/test-network/compose/docker/docker-compose-bft-test-net.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/compose/compose-couch.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/compose/docker/docker-compose-couch.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/compose/compose-ca.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/compose/docker/docker-compose-ca.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/addOrg3/compose/compose-org3.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/addOrg3/compose/docker/docker-compose-org3.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/addOrg3/compose/compose-couch-org3.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/addOrg3/compose/compose-ca-org3.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/rkprac1/Fabric-samples/test-network/addOrg3/compose/docker/docker-compose-ca-org3.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[*] Running 11/11
✓ Container peer0.org2.example.com    Removed          0.8s
✓ Container orderer.example.com       Removed          0.6s
✓ Container peer0.org1.example.com    Removed          0.9s
✓ Volume compose_peer0.org3.example.com Removed          0.0s
✓ Volume compose_orderer4.example.com Removed          0.0s
✓ Volume compose_peer0.org1.example.com Removed          0.0s
✓ Volume compose_peer0.org2.example.com Removed          0.0s
✓ Volume compose_orderer.example.com  Removed          0.0s
✓ Volume compose_orderer2.example.com Removed          0.0s
```

Practical 3

Aim: Interact with a block chain network. Execute transactions and requests against a block chain network by creating an app to test the network and its rules.

Step: Open Remix IDE (<https://remix.ethereum.org/>)



Step 2: Create a file named prac3.sol and type the code

This screenshot shows the code editor in the Remix IDE with a file named 'prac3.sol' open. The code is a Solidity smart contract with several functions: 'ItemListed', 'getAllItems', 'getItemCount', 'transferItem', and 'getItemsByAccount'. The code uses comments to explain its purpose and includes require statements for item existence and owner validation.

```

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract Marketplace {

    struct Item {
        uint id;
        string name;
        uint price;
        address payable seller;
        address owner;
        bool sold;
    }

    uint public itemCount;
    mapping(uint => Item) public items;
    mapping(address => uint[]) public ownerItems;

    event ItemListed(uint itemId, string name, uint price, address seller);
    event ItemPurchased(uint itemId, address buyer, uint price);
    event ItemTransferred(uint itemId, address from, address to);

    // Function to list an item for sale
    function listItem(string memory _name, uint _price) public {
        require(_price > 0, "Price must be greater than zero");

        itemCount++;
        items[itemCount] = Item(

```

```

        itemCount,
        _name,
        _price,
        payable(msg.sender),
        msg.sender,
        false
    );

    ownerItems[msg.sender].push(itemCount);

    emit ItemListed(itemCount, _name, _price, msg.sender);
}

// Function to show all listed items

function getAllItems() public view returns (Item[] memory) {
    Item[] memory allItems = new Item[](itemCount);
    for (uint i = 1; i <= itemCount; i++) {
        allItems[i - 1] = items[i];
    }
    return allItems;
}

// Function to get item count

function getItemCount() public view returns (uint) {
    return itemCount;
}

// Function to transfer an item to another account

```

```

function transferItem(uint _itemId, address _to) public {
    require(_itemId > 0 && _itemId <= itemCount, "Item does not exist");
    require(items[_itemId].owner == msg.sender, "You do not own this item");
    require(_to != address(0), "Invalid address");

    // Remove the item from current owner list
    _removeItemFromOwner(msg.sender, _itemId);

    // Update item owner
    items[_itemId].owner = _to;
    ownerItems[_to].push(_itemId);

    emit ItemTransferred(_itemId, msg.sender, _to);
}

// Function to list all accounts and their items
function getItemsByAccount(address _account) public view returns (Item[] memory) {
    uint[] memory accountItemIds = ownerItems[_account];
    Item[] memory accountItems = new Item[](accountItemIds.length);

    for (uint i = 0; i < accountItemIds.length; i++) {
        accountItems[i] = items[accountItemIds[i]];
    }

    return accountItems;
}

// Internal function to remove an item from the owner's list
function _removeItemFromOwner(address _owner, uint _itemId) internal {

```

```

        uint[] storage itemList = ownerItems[_owner];

        for (uint i = 0; i < itemList.length; i++) {
            if (itemList[i] == _itemId) {
                itemList[i] = itemList[itemList.length - 1]; // Move the last element to the current position
                itemList.pop(); // Remove the last element
                break;
            }
        }

    }

// Function to get item details

function getItem(uint _itemId) public view returns (
    string memory name,
    uint price,
    address seller,
    address owner,
    bool sold
) {
    require(_itemId > 0 && _itemId <= itemCount, "Item does not exist");
    Item storage item = items[_itemId];
    return (item.name, item.price, item.seller, item.owner, item.sold);
}
}

```

Step 3: Click on deploy and run

The screenshot shows the deployment of the Marketplace contract. The sidebar lists functions: listItem, transferItem, getAllItems, getItem, getItemCount, getItemsByAc..., itemCount, Items, and ownerItems. The code editor contains the constructor logic:

```

53     return itemCount;
54 }
55
56 // Function to transfer an item to another account
57 function transferItem(uint _itemId, address _to) public {
58     require(_itemId > 0 && _itemId <= itemCount, "Item does not exist");
59     require(items[_itemId].owner == msg.sender, "You do not own this item");
60     require(_to != address(0), "Invalid address");

```

The terminal shows the deployment command and its output:

```

[vm] from: 0x583...eddC4 to: Marketplace.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x026...3e697

```

Add list item:

The screenshot shows the listItem transaction. The sidebar lists functions: listItem, transferItem, getAllItems, getItem, getItemCount, getItemsByAc..., itemCount, Items, and ownerItems. The code editor contains the listItem logic:

```

53     return itemCount;
54 }
55
56 // Function to transfer an item to another account
57 function transferItem(uint _itemId, address _to) public {
58     require(_itemId > 0 && _itemId <= itemCount, "Item does not exist");
59     require(items[_itemId].owner == msg.sender, "You do not own this item");
60     require(_to != address(0), "Invalid address");

```

The terminal shows the listItem transaction logs:

```

[vm] from: 0x583...eddC4 to: Marketplace.listItem(string,uint256) 0x7b9...b6AcE value: 0 wei data: 0x851...00000 logs: 1 hash: 0xffa...8f84d
[vm] from: 0x583...eddC4 to: Marketplace.listItem(string,uint256) 0x7b9...b6AcE value: 0 wei data: 0x851...00000 logs: 1 hash: 0x908...3371d
[vm] from: 0x583...eddC4 to: Marketplace.listItem(string,uint256) 0x7b9...b6AcE value: 0 wei data: 0x851...00000 logs: 1 hash: 0x452...25570

```

Get all items:

The screenshot shows the getAllItems transaction. The sidebar lists functions: listItem, transferItem, getAllItems, getItem, getItemCount, getItemsByAc..., itemCount, Items, and ownerItems. The code editor contains the getAllItems logic:

```

53     return itemCount;
54 }
55
56 // Function to transfer an item to another account
57 function transferItem(uint _itemId, address _to) public {
58     require(_itemId > 0 && _itemId <= itemCount, "Item does not exist");
59     require(items[_itemId].owner == msg.sender, "You do not own this item");
60     require(_to != address(0), "Invalid address");

```

The terminal shows the getAllItems transaction logs:

```

[vm] from: 0x583...eddC4 to: Marketplace.listItem(string,uint256) 0x7b9...b6AcE value: 0 wei data: 0x851...00000 logs: 1 hash: 0x908...3371d
[vm] from: 0x583...eddC4 to: Marketplace.listItem(string,uint256) 0x7b9...b6AcE value: 0 wei data: 0x851...00000 logs: 1 hash: 0x452...25570
[call] from: 0x58380a6a701c568545dcfc003fc8875f56beddC4 to: Marketplace.getAllItems() data: 0x4ba...1d6aa
[call] from: 0x58380a6a701c568545dcfc003fc8875f56beddC4 to: Marketplace.getItemCount() data: 0x774...9cf23
[call] from: 0x58380a6a701c568545dcfc003fc8875f56beddC4 to: Marketplace.itemCount() data: 0x6bf...b0d01

```

View item details:

```

getitem 1
0: string: name Tissot 1853
1: uint256: price 1
2: address: seller 0x5B38Da6a701c568545dCfcB03
FcB875f56beddC4
3: address: owner 0xB5B38Da6a701c568545dCfcB03
FcB875f56beddC4
4: bool: sold false

getItemCount
0: uint256: 3

getItemsByAc... address _account
itemCount
0: uint256: 3

items
0: uint256: id 3
1: string: name Iphone 18
2: uint256: price 3
3: address: seller 0x5B38Da6a701c568545dCfcB03
FcB875f56beddC4
4: address: owner 0xB5B38Da6a701c568545dCfcB03
FcB875f56beddC4
5: bool: sold false
  
```

Transfer item and get all transferred items via wallet id: 1, 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

```

transaction 2.0Ab8483F64d9C6d1EcF9b849Ae677dD3315835cb2
getItems
0: tuple(uint256,string,uint256,address,address,bo
ol)[]
1: string: name Tissot 1853
2: uint256: price 1
3: address: seller 0xB5B38Da6a701c568545dCfcB03
FcB875f56beddC4
4: address: owner 0xB5B38Da6a701c568545dCfcB03
FcB875f56beddC4
5: bool: sold false

getitem 1
0: string: name Tissot 1853
1: uint256: price 1
2: address: seller 0xB5B38Da6a701c568545dCfcB03
FcB875f56beddC4
3: address: owner 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
4: bool: sold false

getItemCount
0: uint256: 3

getItemsByAc... address _account
itemCount
0: uint256: 3

items
0: uint256: id 3
1: string: name Iphone 18
2: uint256: price 3
3: address: seller 0x5B38Da6a701c568545dCfcB03
FcB875f56beddC4
4: address: owner 0xB5B38Da6a701c568545dCfcB03
FcB875f56beddC4
5: bool: sold false
  
```

```

getItemsByAc...
0x617F2E2fD72FD9D5503197092a
0: tuple(uint256,string,uint256,address,address,bo
ol)[]
1: string: name Razorpay rez
2: uint256: price 2
3: address: seller 0x5B38Da6a701c568545dCfcB03
FcB875f56beddC4
4: address: owner 0xAb8483F64d9C6d1EcF9b849A
e677dD3315835cb2
5: bool: sold false
  
```

Now you can check seller and new owner

```

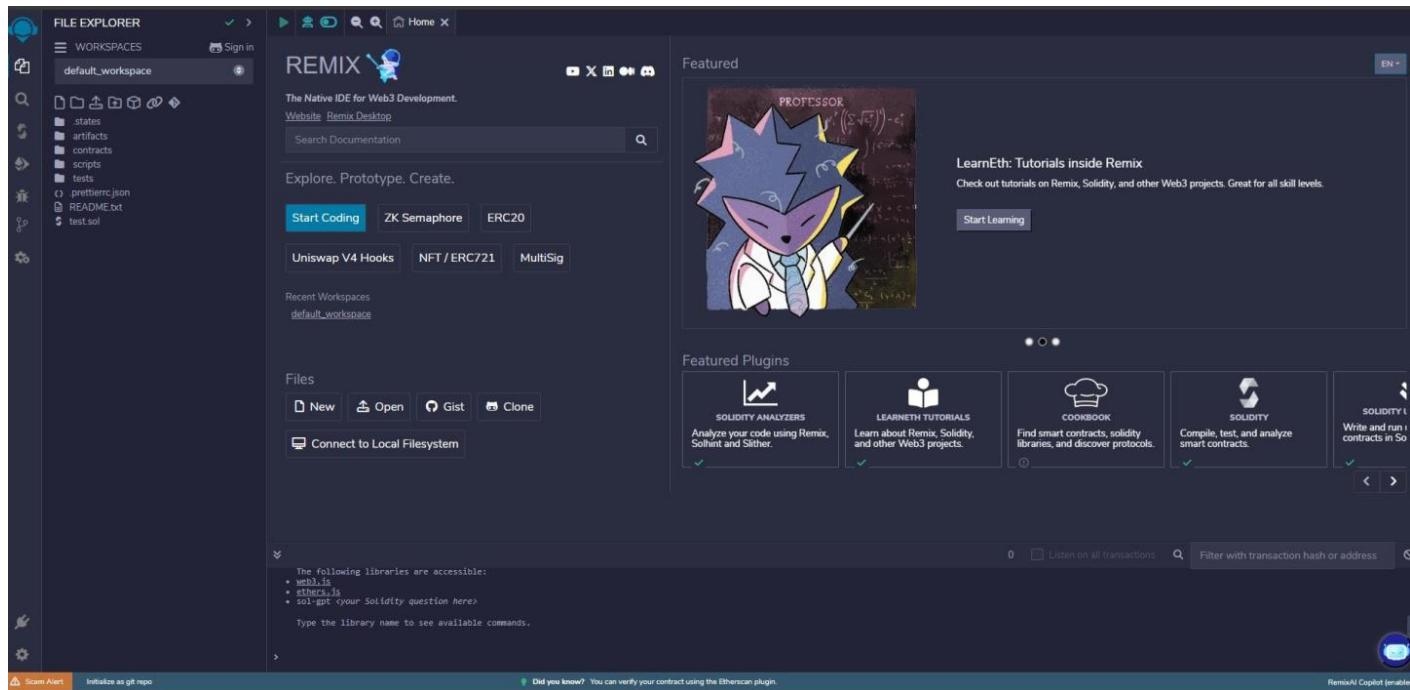
0: uint256: 3

items
0: uint256: id 2
1: string: name Razorpay rez
2: uint256: price 2
3: address: seller 0x5B38Da6a701c568545dCfcB03
FcB875f56beddC4
4: address: owner 0xAb8483F64d9C6d1EcF9b849A
e677dD3315835cb2
5: bool: sold false
  
```

Practical 4

Aim: Deploy an asset-transfer app using block chain. Learn app development within a Hyperledger Fabric network.

Step: Open Remix IDE (<https://remix.ethereum.org/>)



Step 2: Create a file named prac4.sol and type the code

This screenshot shows the Remix IDE with the file 'prac4.sol' selected in the File Explorer. The code defines a Solidity contract named 'Asset'. It includes variables for owner count, contract owner, land ID, land square feet, and creation date. The constructor sets the initial owner. The 'createLand' function creates a new land entry with the provided parameters. The 'onlyOwner' modifier ensures only the contract owner can perform certain actions. The 'transferOwnership' function allows the owner to change the contract owner to a new address. The 'previousOwner' function returns the address of the previous owner if there is one, otherwise returns the zero address. The code uses standard Solidity syntax with comments explaining its functionality.

```

// SPDX-License-Identifier: UNLICENSED

pragma solidity ^0.8.7;

contract Asset {

    uint256 public owners_count;
    address public contract_owner;
    uint256 public land_id;
    uint256 public land_sqrfeet;
    uint256 public land_created_date;
    mapping(uint256 => address) public owners;

    constructor() {
        contract_owner = msg.sender; // Set the deployer as the initial owner
    }

    function createland(uint256 _land_id, uint256 _land_sqrfeet, uint256 _land_created_date) public returns (bool){
        setOwner(msg.sender);
        land_id = _land_id;
        land_sqrfeet = _land_sqrfeet;
        land_created_date = _land_created_date;
        return true;
    }

    modifier onlyOwner(){
        require(msg.sender == contract_owner, "Only the contract owner can perform this action");
        _;
    }
}

```

```

function transferOwnership(address _newOwner) public onlyOwner {
    require(_newOwner != address(0), "New owner address cannot be zero");
    contract_owner = _newOwner;
}

function previousOwner() view public returns(address) {
    if (owners_count > 0) {
        uint256 previous_owner = owners_count - 1;
        return owners[previous_owner];
    }
    return address(0); // Return a default value (e.g., zero address) if no previous owner exists
}

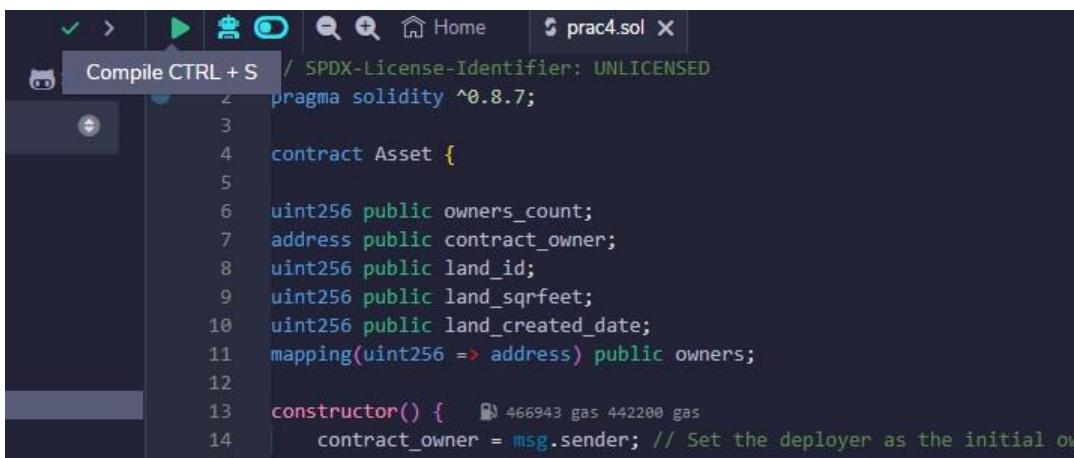
function setOwner(address owner) public {
    owners_count += 1;
    owners[owners_count] = owner;
}

function getCurrentOwner() view public returns(address){
    return owners[owners_count];
}

function getOwnerCount() view public returns(uint256){
    return owners_count;
}

```

Step 4: Click on compile

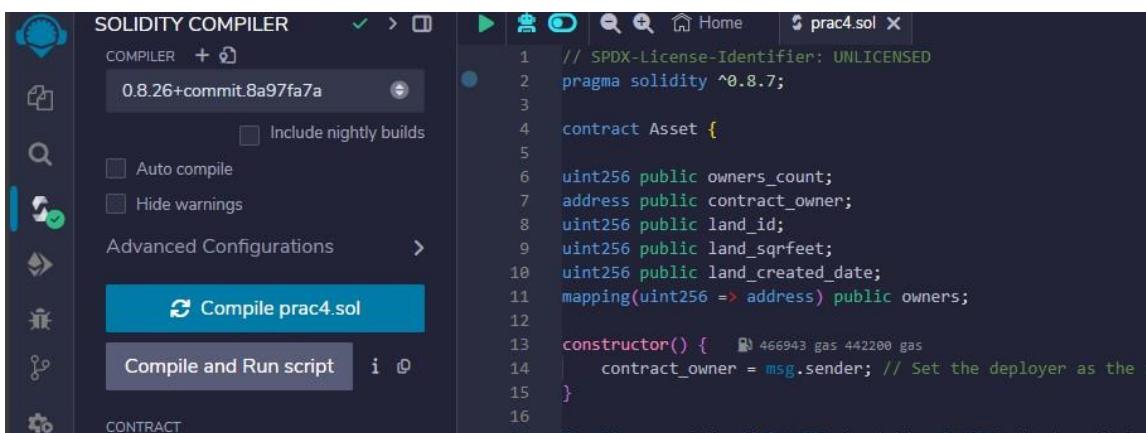


```
pragma solidity ^0.8.7;

contract Asset {
    uint256 public owners_count;
    address public contract_owner;
    uint256 public land_id;
    uint256 public land_sqfeet;
    uint256 public land_created_date;
    mapping(uint256 => address) public owners;

    constructor() {
        contract_owner = msg.sender; // Set the deployer as the initial owner
    }
}
```

Step 5: on left sided panel it will show success then click run and deploy



SOLIDITY COMPILER

COMPILER + 0.8.26+commit.8a97fa7a

Include nightly builds

Auto compile

Hide warnings

Advanced Configurations >

Compile prac4.sol

Compile and Run script

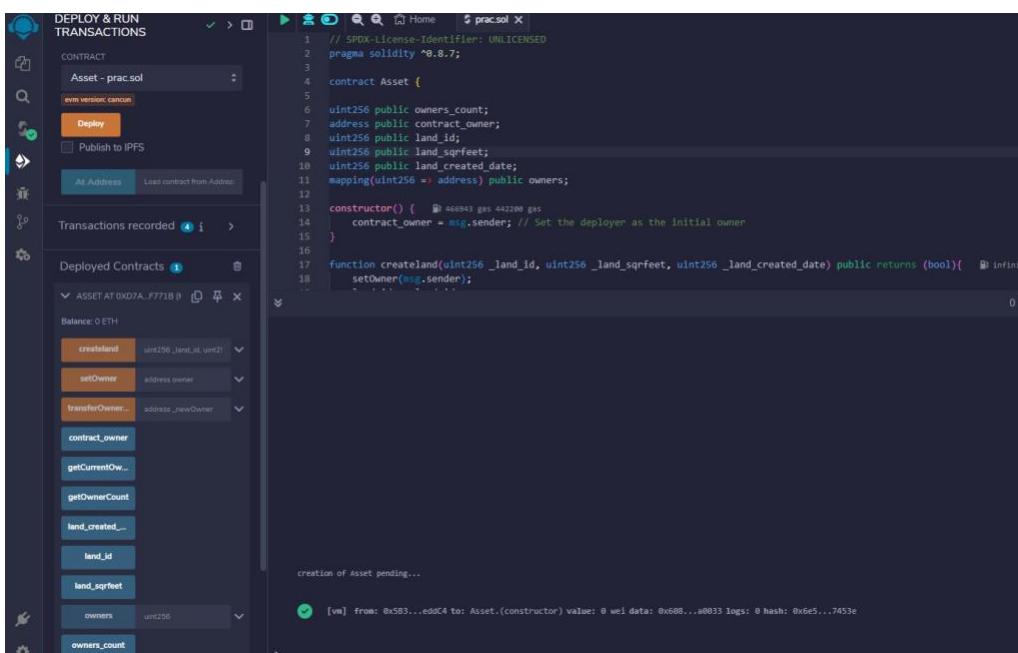
CONTRACT

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.7;

contract Asset {
    uint256 public owners_count;
    address public contract_owner;
    uint256 public land_id;
    uint256 public land_sqfeet;
    uint256 public land_created_date;
    mapping(uint256 => address) public owners;

    constructor() {
        contract_owner = msg.sender; // Set the deployer as the initial owner
    }
}
```

Step 6: Select Environment and click deploy



DEPLOY & RUN TRANSACTIONS

CONTRACT Asset - prac.sol

Deploy

Publish to IPFS

At Address Load contract from Address

Transactions recorded 0

Deployed Contracts 0

ASSET AT 0xD7A...F771B0

Balance: 0 ETH

cresteland uint256 _land_id, uint256 _land_sqfeet

setOwner address owner

transferOwner... address _newOwner

contract_owner

getCurrentOwn...

getOwnerCount

land_created...

land_id

land_sqfeet

owners uint256

owners_count

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.7;

contract Asset {
    uint256 public owners_count;
    address public contract_owner;
    uint256 public land_id;
    uint256 public land_sqfeet;
    uint256 public land_created_date;
    mapping(uint256 => address) public owners;

    constructor() {
        contract_owner = msg.sender; // Set the deployer as the initial owner
    }

    function createland(uint256 _land_id, uint256 _land_sqfeet, uint256 _land_created_date) public returns (bool) {
        setOwner(msg.sender);
    }
}
```

creation of Asset pending...

[vm] from: 0x5B3...eddC4 to: Asset.(constructor) value: 0 wei data: 0x688...a0833 logs: 0 hash: 0x6e5...7453e

After deploying you'll get the console success and on left createland, setowner and transferown option will come where you have to add the value.

CreateLand : 12345678901234567890,150000,1672531200

After adding the values you'll get the success

The screenshot shows the Truffle UI interface for the Asset contract. On the left, there is a sidebar with several buttons: 'createland' (orange), 'setOwner' (orange), 'transferOwner...' (orange), 'contract_owner' (blue), 'getCurrentOw...', 'getOwnerCount' (blue), 'land_created_...', 'land_id' (blue), 'land_sqfeet' (blue), 'owners' (blue), 'owners_count' (blue), and 'previousOwner' (blue). Below these are sections for 'Low level interactions' and 'CALLDATA'. A 'Transact' button is located at the bottom of the sidebar. The main area displays the Solidity code for the Asset contract. At the bottom right, there is a status bar with '0' transactions and a link to 'Listen on all transactions'.

```
9 uint256 public land_sqfeet;
10 uint256 public land_created_date;
11 mapping(uint256 => address) public owners;
12
13 constructor() {
14     contract_owner = msg.sender; // Set the deployer as the initial owner
15 }
16
17 function createland(uint256 _land_id, uint256 _land_sqfeet, uint256 _land_created_date) public returns (bool){ infinite gas
18     setOwner(msg.sender);
}
```

creation of Asset pending...

- [vm] from: 0x5B3...eddC4 to: Asset.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0xe5...7453e transact to Asset.createland pending ...
- [vm] from: 0x5B3...eddC4 to: Asset.createland(uint256,uint256,uint256) 0x07A...F771B value: 0 wei data: 0x5c7...0cd00 logs: 0 hash: 0x989...df8b7 transact to Asset.setOwner pending ...
- [vm] from: 0x5B3...eddC4 to: Asset.setOwner(address) 0x07A...F771B value: 0 wei data: 0x13a...eddc4 logs: 0 hash: 0x8fa...b828c transact to Asset.transferOwnership pending ...
- [vm] from: 0x5B3...eddC4 to: Asset.transferOwnership(address) 0x07A...F771B value: 0 wei data: 0xf2f...c02db logs: 0 hash: 0x07c...d6861

On clicking the blue buttons you can see the results

This screenshot shows the same Truffle UI interface as the previous one, but it has been deployed. The sidebar buttons are now greyed out. The main area displays the same Solidity code as before. The transaction history at the bottom is identical to the previous screenshot, showing the creation of the asset and the transfer of ownership.

```
7 address public contract_owner;
8 uint256 public land_id;
9 uint256 public land_sqfeet;
10 uint256 public land_created_date;
11 mapping(uint256 => address) public owners;
12
13 constructor() {
14     contract_owner = msg.sender; // Set the deployer as the initial owner
15 }
16
17 function createland(uint256 _land_id, uint256 _land_sqfeet, uint256 _land_created_date) public returns (bool){ infinite gas
18     setOwner(msg.sender);
}
```

CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: Asset.getOwnerCount() data: 0xef1...8374a call to Asset.land_created_date

CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: Asset.land_created_date() data: 0x97d...b9948 call to Asset.land_id

CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: Asset.land_id() data: 0xa6d...c737a call to Asset.land_sqfeet

CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: Asset.land_sqfeet() data: 0x2e3...6487c call to Asset.owners_count

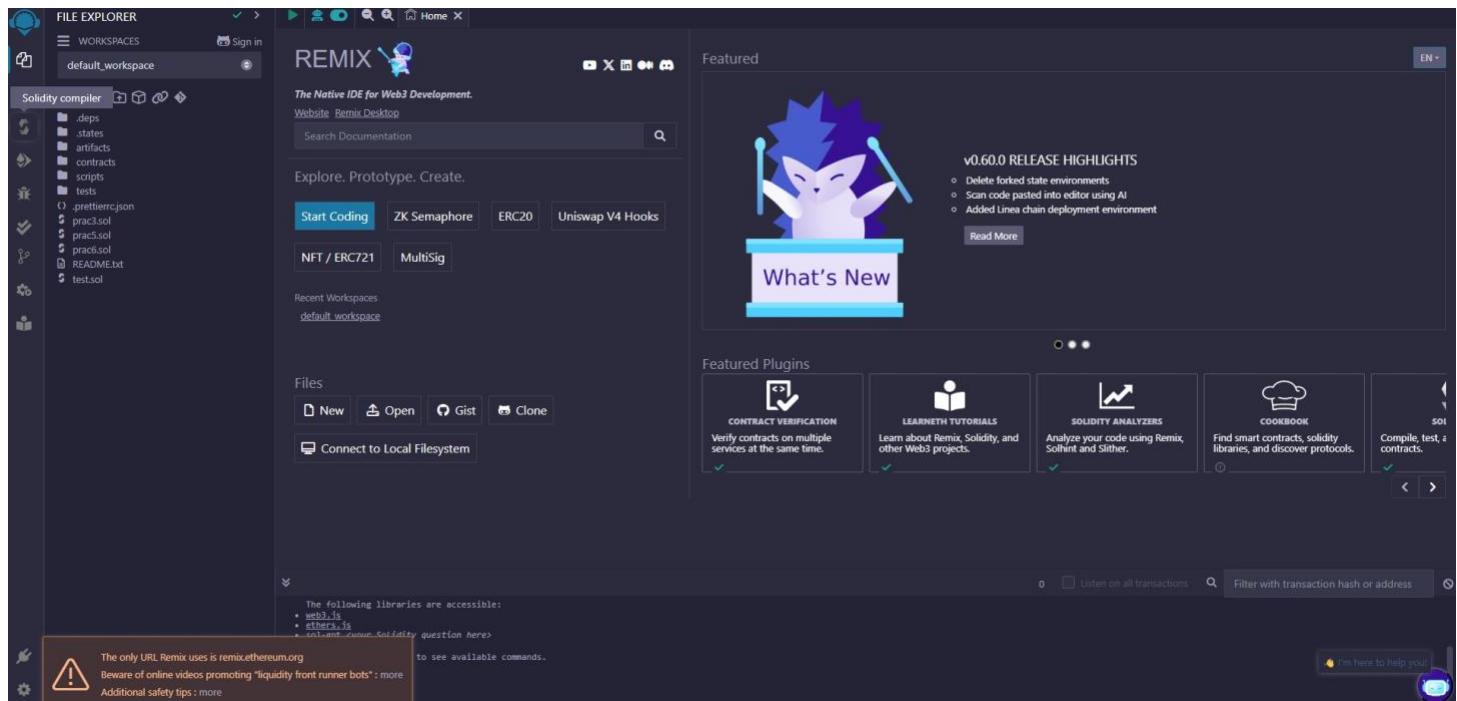
CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: Asset.owners_count() data: 0x59c...42e15 call to Asset.previousOwner

CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: Asset.previousOwner() data: 0x674...f220f

Practical 5

Aim: Interact with a block chain network. Execute transactions and requests against a block chain network by creating an app to test the network and its rules.

Step: Open Remix IDE (<https://remix.ethereum.org/>)



Step 2: Create a file named prac3.sol and type the code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract FitnessClubRewards {
    // Define a structure for tracking each user's activity and rewards
    struct Member {
        uint256 totalWorkouts; // Total number of workouts by the user
        uint256 rewardPoints; // Rewards points based on workouts
        bool isMember; // Check if user is a registered member
    }

    // Mapping to store members by their Ethereum address
    mapping(address => Member) public members;

    // Event for logging new member registration
    event MemberRegistered(address indexed user);

    // Event for logging workout completion
    event WorkoutCompleted(address indexed user, uint256 newTotalWorkouts, uint256 rewardPointsEarned);

    // Event for redeeming reward points
    event RewardPointsRedeemed(address indexed user, uint256 pointsRedeemed);

    // Register a new member (can be done by the club admin or through a special function)
    function registerMember(address user) public {
        require(!members[user].isMember, "User is already a member");

        members[user] = Member({
            totalWorkouts: 0,
            rewardPoints: 0,
            isMember: true
        });

        emit MemberRegistered(user);
    }
}
```

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract FitnessClubRewards {

    // Define a structure for tracking each user's activity and rewards
    struct Member {
        uint256 totalWorkouts; // Total number of workouts by the user
        uint256 rewardPoints; // Rewards points based on workouts
        bool isMember; // Check if user is a registered member
    }

    // Mapping to store members by their Ethereum address
    mapping(address => Member) public members;

    // Event for logging new member registration
    event MemberRegistered(address indexed user);

    // Event for logging workout completion
    event WorkoutCompleted(address indexed user, uint256 newTotalWorkouts, uint256 rewardPointsEarned);

    // Event for redeeming reward points
    event RewardPointsRedeemed(address indexed user, uint256 pointsRedeemed);

    // Register a new member (can be done by the club admin or through a special function)
    function registerMember(address user) public {
        require(!members[user].isMember, "User is already a member");
    }
}
```

```

members[user] = Member({
    totalWorkouts: 0,
    rewardPoints: 0,
    isMember: true
});

emit MemberRegistered(user);

}

// Record a workout completion for a member
function completeWorkout(address user) public {
    require(members[user].isMember, "User is not a member");

    // Increment the total workouts by 1 and update the reward points
    members[user].totalWorkouts++;

    // Every workout gives 10 reward points (this can be customized)
    members[user].rewardPoints += 10;

    emit WorkoutCompleted(user, members[user].totalWorkouts, 10);
}

// Redeem reward points for the user
function redeemPoints(address user, uint256 points) public {
    require(members[user].isMember, "User is not a member");
    require(members[user].rewardPoints >= points, "Not enough points");

    // Deduct the points from the user's balance
}

```

```

members[user].rewardPoints -= points;

emit RewardPointsRedeemed(user, points);

}

// Get user's current rewards balance

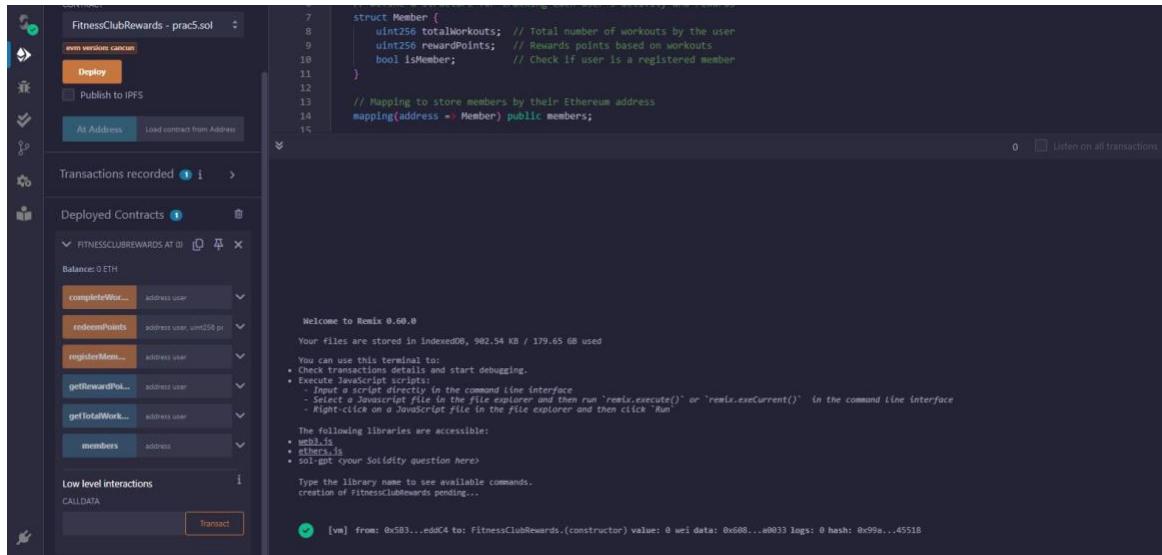
function getRewardPoints(address user) public view returns (uint256) {
    require(members[user].isMember, "User is not a member");
    return members[user].rewardPoints;
}

// Get user's total workouts

function getTotalWorkouts(address user) public view returns (uint256) {
    require(members[user].isMember, "User is not a member");
    return members[user].totalWorkouts;
}

```

Now compile and then run and deploy



Now you can register, complete and redeem the points

The screenshot shows the Remix IDE interface for the FitnessClubRewards contract. The sidebar lists several functions:

- completeWorkout: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2
- redeemPoints: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2, 10
- registerMember: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2
- getRewardPoints: address user
- getTotalWorkouts: address user
- members: address

Under "Low level interactions", there is a "CALLDATA" section and a "Transact" button.

The right side displays the transaction logs:

- [vm] from: 0x5B3...edd4 to: FitnessClubRewards.registerMember(address) 0x5A8...C4d01 value: 0 wei data: 0xfc...35cb2 logs: 1 hash: 0xb3...96e6 transact to FitnessClubRewards.registerMember pending ...
- [vm] from: 0x5B3...edd4 to: FitnessClubRewards.completeWorkout(address) 0x5A8...C4d01 value: 0 wei data: 0x82c...35cb2 logs: 1 hash: 0xc2...1dc1 transact to FitnessClubRewards.completeWorkout pending ...
- [vm] from: 0x5B3...edd4 to: FitnessClubRewards.redeemPoints(address, uint256) 0x5A8...C4d01 value: 0 wei data: 0xb85...75cb0 logs: 0 hash: 0x3c...e14a transact to FitnessClubRewards.redeemPoints errored: Error occurred: revert. revert The transaction has been reverted to the initial state. Reason provided by the contract: "Not enough points". If the transaction failed for not having enough gas, try increasing the gas limit gently. transact to FitnessClubRewards.redeemPoints pending ...
- [vm] from: 0x5B3...edd4 to: FitnessClubRewards.redeemPoints(address, uint256) 0x5A8...C4d01 value: 0 wei data: 0xb85...0000a logs: 1 hash: 0x541...23e6d

Now you can total members, getreward, total workout summary

The screenshot shows the Remix IDE interface for the FitnessClubRewards contract. The sidebar lists several functions:

- getRewardPoints: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2
- getTotalWorkouts: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2
- members: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2

Under "Low level interactions", there is a "CALLDATA" section and a "Transact" button.

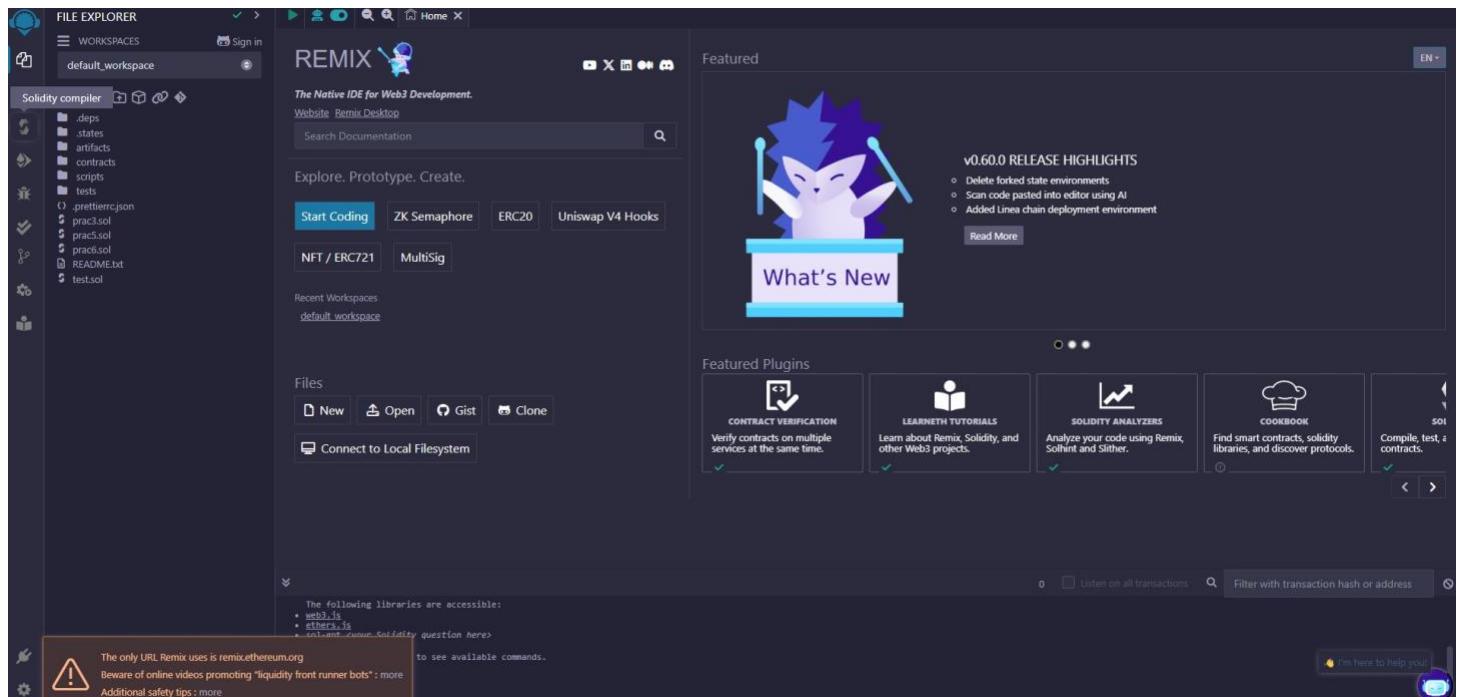
The right side displays the transaction logs:

- [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2 to: FitnessClubRewards.completeWorkout pending ...
- [vm] from: 0xA08...35cb2 to: FitnessClubRewards.completeWorkout(address) 0x5A8...C4d01 value: 0 wei data: 0x82c...35cb2 logs: 1 hash: 0xc0...698e call to FitnessClubRewards.getRewardPoints
- [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2 to: FitnessClubRewards.getRewardPoints(address) data: 0xfd6...35cb2
- [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2 to: FitnessClubRewards.getTotalWorkouts data: 0x84f...35cb2
- [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2 to: FitnessClubRewards.members data: 0x80a...35cb2

Practical 6

Aim: Interact with a block chain network. Execute transactions and requests against a block chain network by creating an app to test the network and its rules.

Step: Open Remix IDE (<https://remix.ethereum.org/>)



Step 2: Create a file named prac6.sol and type the code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract MemberRewards {
    struct Member {
        uint points;
        string name;
    }

    // Mapping to store member rewards by their address
    mapping(address => Member) public members;

    // Event to log reward changes
    event RewardUpdated(address indexed member, uint newPoints);

    // Function to add a member (could be restricted to admins)
    function addMember(address memberAddress, string memory memberName) public {
        members[memberAddress] = Member(0, memberName); // Initialize with 0 points
    }

    // Function to get member's current points
    function getReward(address memberAddress) public view returns (uint) {
        return members[memberAddress].points;
    }

    // Function to add reward points to a member
    function addReward(address memberAddress, uint points) public {
        require(members[memberAddress].points >= 0, "Member does not exist");
        members[memberAddress].points += points;
        emit RewardUpdated(memberAddress, members[memberAddress].points);
    }

    // Function to subtract reward points from a member (e.g., for redemption)
    function subtractReward(address memberAddress, uint points) public {
        require(members[memberAddress].points >= points, "Insufficient points");
        members[memberAddress].points -= points;
    }
}
```

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract MemberRewards {

    struct Member {
        uint points;
        string name;
    }

    // Mapping to store member rewards by their address
    mapping(address => Member) public members;

    // Event to log reward changes
    event RewardUpdated(address indexed member, uint newPoints);

    // Function to add a member (could be restricted to admins)
    function addMember(address memberAddress, string memory memberName) public {
        members[memberAddress] = Member(0, memberName); // Initialize with 0 points
    }

    // Function to get member's current points
    function getReward(address memberAddress) public view returns (uint) {
        return members[memberAddress].points;
    }

    // Function to add reward points to a member
    function addReward(address memberAddress, uint points) public {
```

```

require(members[memberAddress].points >= 0, "Member does not exist");

members[memberAddress].points += points;

emit RewardUpdated(memberAddress, members[memberAddress].points);

}

// Function to subtract reward points from a member (e.g., for redemption)

function subtractReward(address memberAddress, uint points) public {

    require(members[memberAddress].points >= points, "Insufficient points");

    members[memberAddress].points -= points;

    emit RewardUpdated(memberAddress, members[memberAddress].points);

}

```

Step 3: Now compile and then Deploy and run

The screenshot shows the Truffle UI interface for the MemberRewards contract. On the left, there's a sidebar with the contract name 'MemberRewards - prac6.sol', the EVM version 'cancun', and a 'Deploy' button. Below that are sections for 'Transactions recorded' and 'Deployed Contracts'. Under 'Deployed Contracts', the 'MEMBERREWARDS AT 0x38C...24CT3 (MEMORY)' is listed with a balance of 0 ETH. It shows several functions: 'addMember', 'addReward', 'subtractReward', 'getReward', and 'members'. The right side of the interface displays the full Solidity code for the contract, including comments and function definitions. At the bottom, a terminal window shows the deployment command and its output, indicating a successful deployment.

```

MemberRewards - prac6.sol
evm version: cancan
Deploy
Publish to IPFS
At Address Load contract from Address

Transactions recorded 1 i
Deployed Contracts 1
MEMBERREWARDS AT 0x38C...24CT3 (MEMORY)
Balance: 0 ETH
addMember address: MemberAddress, string: memberName
addReward address: MemberAddress, uint256: points
subtractReward address: MemberAddress, uint256: points
getReward address: MemberAddress
members address

Low level interactions i
CALLDATA
Transact

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

}

// Mapping to store member rewards by their address
mapping(address => Member) public members;

// Event to log reward changes
event RewardUpdated(address indexed member, uint newPoints);

// Function to add a member (could be restricted to admins)
function addMember(address memberAddress, string memory memberName) public {
    members[memberAddress] = Member(0, memberName); // Initialize with 0 points
}

// Function to get member's current points
function getReward(address memberAddress) public view returns (uint) {
    return members[memberAddress].points;
}

// Function to add reward points to a member
function addReward(address memberAddress, uint points) public {
    require(members[memberAddress].points >= 0, "Member does not exist");
    members[memberAddress].points += points;
    emit RewardUpdated(memberAddress, members[memberAddress].points);
}

// Function to subtract reward points from a member (e.g., for redemption)
function subtractReward(address memberAddress, uint points) public {
    require(members[memberAddress].points >= points, "Insufficient points");
    members[memberAddress].points -= points;
}

Type the library name to see available commands.
creation of MemberRewards pending...
[vm] from: 0x5B3...eddC4 to: MemberRewards.(constructor) value: 0 wei data: 0x600...a0033 logs: 0 hash: 0x772...7f948

```

Here you can add a member, add points and even get the summary of the reward and members

```

25     }
26
27     // Function to add reward points to a member
28     function addReward(address memberAddress, uint points) public {
29         require(members[memberAddress].points >= 0, "Member does not exist");
30         members[memberAddress].points += points;
31         emit RewardUpdated(memberAddress, members[memberAddress].points);
32     }
33
34     // Function to subtract reward points from a member (e.g., for redemption)
35     function subtractReward(address memberAddress, uint points) public {
36         require(members[memberAddress].points >= points, "Insufficient points");
37         members[memberAddress].points -= points;
38     }

```

call to MemberRewards.members

call [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: MemberRewards.members(address) data: 0x08a...35cb2

Subtract the reward

```

33     // Function to subtract reward points from a member (e.g., for redemption)
34     function subtractReward(address memberAddress, uint points) public {
35         require(members[memberAddress].points >= points, "Insufficient points");
36         members[memberAddress].points -= points;
37     }

```

call to MemberRewards.members

call [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: MemberRewards.members(address) data: 0x08a...35cb2

You can add multiple members and manage it

```

22     // Function to get member's current points
23     function getReward(address memberAddress) public view returns (uint) {
24         return members[memberAddress].points;
25     }
26
27     // Function to add reward points to a member
28     function addReward(address memberAddress, uint points) public {
29         require(members[memberAddress].points >= 0, "Member does not exist");
30         members[memberAddress].points += points;
31         emit RewardUpdated(memberAddress, members[memberAddress].points);
32     }
33
34     // Function to subtract reward points from a member (e.g., for redemption)
35     function subtractReward(address memberAddress, uint points) public {
36         require(members[memberAddress].points >= points, "Insufficient points");
37         members[memberAddress].points -= points;
38     }

```

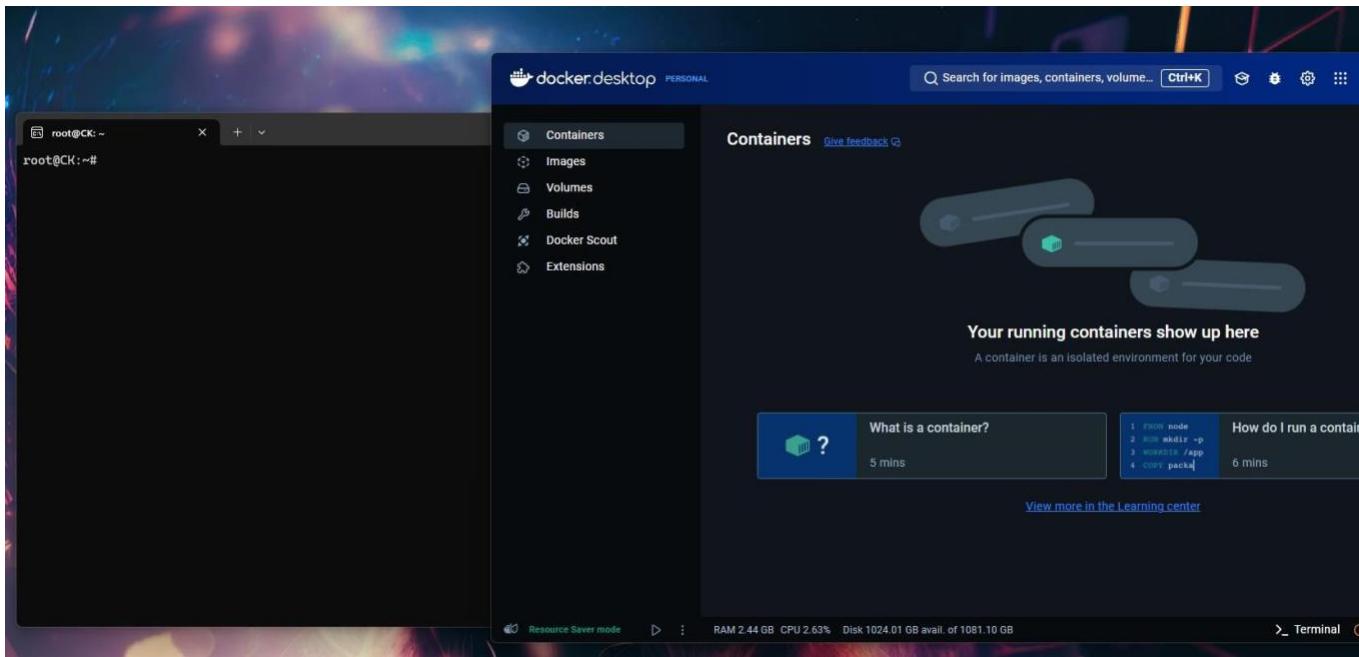
call to MemberRewards.members

call [call] from: 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabAB to: MemberRewards.members(address) data: 0x08a...cabAB

Practical 7

Aim: Car auction network: A Hello World example with Hyperledger Fabric Node SDK and IBM Block chain Starter Plan. Use Hyperledger Fabric to invoke chaincode while storing results and data in the starter plan

Open Ubuntu and docker desktop too and create an directory and open it



Open Ubuntu terminal → Go to your fabric-samples directory

Run the following command: curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.4

A screenshot of an Ubuntu terminal window titled 'root@CK: ~/prac7'. The terminal shows the following commands and output:

```
root@CK:~# mkdir prac7
root@CK:~# cd prac7
root@CK:~/prac7# curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.4

Clone hyperledger/fabric-samples repo
====> Cloning hyperledger/fabric-samples repo
Cloning into 'fabric-samples'...
remote: Enumerating objects: 14289, done.
remote: Counting objects: 100% (132/132), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 14289 (delta 69), reused 77 (delta 62), pack-reused 14157 (from 4)
Receiving objects: 100% (14289/14289), 23.38 MiB | 11.58 MiB/s, done.
Resolving deltas: 100% (7889/7889), done.
====> Checking out v1.4.4 of hyperledger/fabric-samples

Pull Hyperledger Fabric binaries
====> Downloading version 1.4.4 platform specific fabric binaries
====> Downloading: https://github.com/hyperledger/fabric/releases/download/v1.4.4/hyperledger-fabric-linux-amd64-1.4.4.t
ar.gz
% Total    % Received % Xferd  Average Speed   Time      Time     Current
          Dload Upload   Total Spent  Left Speed
0       0     0      0      0      0      0      0      0      0
0 79.6M  0  4125     0      0  2418      0  9:35:25  0:00:01  9:35:24 24700
```

```

root@CK:~/prac7          + 
6a8f2b9e77bb: Pull complete
ed5069ad5c3: Pull complete
6e8d04642d0a: Pull complete
Digest: sha256:7e9ae6f0e9be95e300206228c67501a57521aa115369823de4aae592c5e468ec
Status: Downloaded newer image for hyperledger/fabric-ca:1.5.13
docker.io/hyperledger/fabric-ca:1.5.13
==> List out hyperledger docker images
hyperledger/fabric-ca    1.5      145d36e5f8ba  3 months ago  214MB
hyperledger/fabric-ca    1.5.13   145d36e5f8ba  3 months ago  214MB
hyperledger/fabric-ca    latest   145d36e5f8ba  3 months ago  214MB
hyperledger/fabric-peer  2.5      e9702d423cd4  6 months ago  142MB
hyperledger/fabric-peer  2.5.9   e9702d423cd4  6 months ago  142MB
hyperledger/fabric-orderer 2.5     10fb520e9b0a  6 months ago  111MB
hyperledger/fabric-orderer 2.5.9   10fb520e9b0a  6 months ago  111MB
hyperledger/fabric-ccenv  2.5     09ef9881ad5f  6 months ago  638MB
hyperledger/fabric-ccenv  2.5.9   09ef9881ad5f  6 months ago  638MB
hyperledger/fabric-baseos 2.5     b6e93e2f93f9  6 months ago  129MB
hyperledger/fabric-baseos 2.5.9   b6e93e2f93f9  6 months ago  129MB
hyperledger/fabric-baseos latest   b6e93e2f93f9  6 months ago  129MB
hyperledger/fabric-ca    1.5.12   e324dc92c6e   6 months ago  209MB
hyperledger/fabric-ccenv 1.4      ca4780293e4c  5 years ago  1.37GB
hyperledger/fabric-ccenv 1.4.4   ca4780293e4c  5 years ago  1.37GB
hyperledger/fabric-ccenv latest   ca4780293e4c  5 years ago  1.37GB
hyperledger/fabric-orderer 1.4     dbc9f65443aa  5 years ago  120MB
hyperledger/fabric-orderer 1.4.4   dbc9f65443aa  5 years ago  120MB
hyperledger/fabric-orderer latest   dbc9f65443aa  5 years ago  120MB
hyperledger/fabric-peer   1.4     9756aed98c6b  5 years ago  128MB
hyperledger/fabric-peer   1.4.4   9756aed98c6b  5 years ago  128MB
hyperledger/fabric-peer   latest   9756aed98c6b  5 years ago  128MB
root@CK:~/prac7#

```

Set up the blockchain network

cd fabric-samples

ls

cd first-network

./byfn.sh down

```

root@CK:~/prac7# cd fabric-samples
root@CK:~/prac7/fabric-samples# ls
CODE_OF_CONDUCT.md  MAINTAINERS.md      basic-network      ci           docs        interest_rate_swaps
CONTRIBUTING.md    README.md          bin                 ci.properties  fabcar      off_chain_data
Jenkinsfile         SECURITY.md       chaincode          commercial-paper first-network scripts
LICENSE            balance-transfer  chaincode-docker-devmode config      high-throughput
root@CK:~/prac7/fabric-samples# cd first-network
root@CK:~/prac7/fabric-samples/first-network# ./byfn.sh down
Stopping for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay of '3' seconds
Continue? [Y/n] Y
proceeding ...
WARN[0000] /root/prac7/fabric-samples/first-network/docker-compose-cli.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/prac7/fabric-samples/first-network/docker-compose-couch.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/prac7/fabric-samples/first-network/docker-compose-kafka.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/prac7/fabric-samples/first-network/docker-compose-etcdraft2.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] The "BYFN_CA2_PRIVATE_KEY" variable is not set. Defaulting to a blank string.
WARN[0000] The "BYFN_CA2_PRIVATE_KEY" variable is not set. Defaulting to a blank string.
WARN[0000] The "BYFN_CAI_PRIVATE_KEY" variable is not set. Defaulting to a blank string.
WARN[0000] The "BYFN_CAI_PRIVATE_KEY" variable is not set. Defaulting to a blank string.
WARN[0000] /root/prac7/fabric-samples/first-network/docker-compose-ca.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] /root/prac7/fabric-samples/first-network/docker-compose-org3.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 11/0
  ✓ Volume net_orderer3.example.com  Removed          0.0s
  ✓ Volume net_peer1.org1.example.com Removed          0.0s
  ✓ Volume net_peer1.org2.example.com Removed          0.0s
  ✓ Volume net_orderer.example.com   Removed          0.0s
  ✓ Volume net_peer0.org3.example.com Removed          0.0s
  ✓ Volume net_orderer5.example.com  Removed          0.0s
  ✓ Volume net_peer1.org3.example.com Removed          0.0s
  ✓ Volume net_orderer2.example.com  Removed          0.0s
  ✓ Volume net_orderer4.example.com  Removed          0.0s
  ✓ Volume net_peer0.org1.example.com Removed          0.0s
Unable to find image 'hyperledger/fabric-tools:latest' locally
latest: Pulling from hyperledger/fabric-tools
002a6d458aa: Downloading [=====] 12.84MB/45.11MB

```

Note, this will take down all of your containers whether they're Fabric related or not.

```
docker rm -f $(docker ps -aq)  
docker rmi -f $(docker images | grep fabcar | awk '{print $3}')
```

Launch the network: Change the directory to fabcar →

```
./startFabric.sh javascript
```

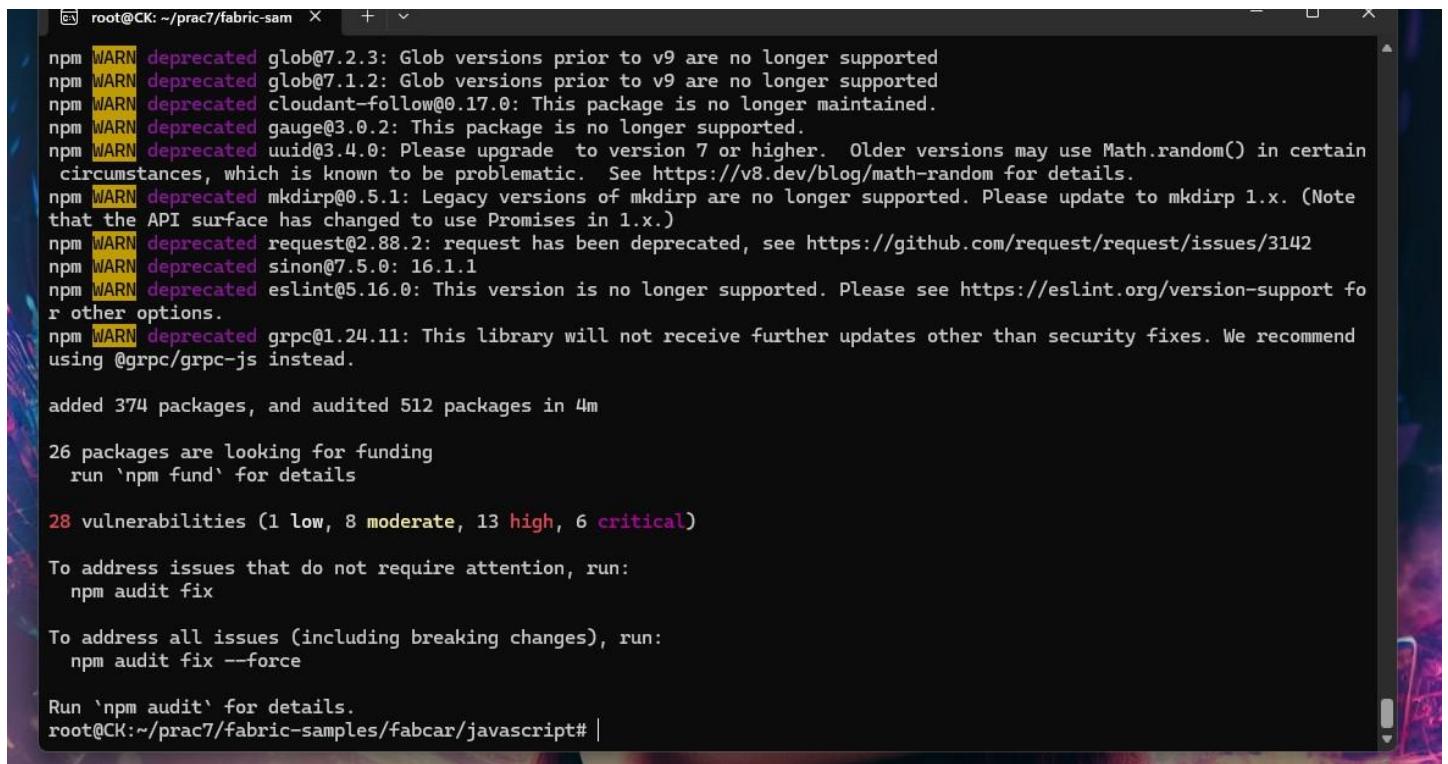
```
Remove one or more images  
root@CK:~/prac7/fabric-samples/first-network# cd ../fabcar  
root@CK:~/prac7/fabric-samples/fabcar# ./startFabric.sh javascript  
Stopping for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay of '3' seconds  
proceeding ...
```

Install the application Run the following command to install the Fabric dependencies for the applications. It will take about a minute to complete:

→ cd javascript

```
sudo apt update  
sudo apt install build-essential python3 python3-pip libssl-dev libz-dev  
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash  
rm -rf node_modules package-lock.json  
npm install
```

→ npm install



A screenshot of a terminal window titled "root@CK: ~/prac7/fabric-samples/fabcar/javascript#". The window displays the output of an "npm audit" command. The output shows numerous warnings and vulnerabilities, including deprecated packages like glob@7.2.3, cloudant-follow@0.17.0, gauge@3.0.2, and uuid@3.4.0. It also mentions legacy versions of mkdirp and request. The audit report indicates 26 packages are looking for funding and lists 28 vulnerabilities across different severity levels (low, moderate, high, critical). It provides instructions for addressing issues, such as running "npm audit fix" for low-severity issues and "npm audit fix --force" for all issues. A note at the bottom says to run "npm audit" for details. The terminal window has a dark background with a colorful decorative border.

```
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported  
npm WARN deprecated glob@7.1.2: Glob versions prior to v9 are no longer supported  
npm WARN deprecated cloudant-follow@0.17.0: This package is no longer maintained.  
npm WARN deprecated gauge@3.0.2: This package is no longer supported.  
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain  
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.  
npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (Note  
that the API surface has changed to use Promises in 1.x.)  
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142  
npm WARN deprecated sinon@7.5.0: 16.1.1  
npm WARN deprecated eslint@5.16.0: This version is no longer supported. Please see https://eslint.org/version-support fo  
r other options.  
npm WARN deprecated grpc@1.24.11: This library will not receive further updates other than security fixes. We recommend  
using @grpc/grpc-js instead.  
  
added 374 packages, and audited 512 packages in 4m  
  
26 packages are looking for funding  
  run 'npm fund' for details  
  
28 vulnerabilities (1 low, 8 moderate, 13 high, 6 critical)  
  
To address issues that do not require attention, run:  
  npm audit fix  
  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
  
Run 'npm audit' for details.  
root@CK:~/prac7/fabric-samples/fabcar/javascript# |
```

Let's enroll user admin:

→ node enrollAdmin.js

```
Run `npm audit` for details.
root@CK:~/prac7/fabric-samples/fabcar/javascript# node enrollAdmin.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Successfully enrolled admin user "admin" and imported it into the wallet
root@CK:~/prac7/fabric-samples/fabcar/javascript#
```

This command has stored the CA administrator's credentials in the wallet directory. Register and enroll node registerUser.js

```
root@CK:~/prac7/fabric-samples/fabcar/javascript# node enrollAdmin.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Successfully enrolled admin user "admin" and imported it into the wallet
root@CK:~/prac7/fabric-samples/fabcar/javascript# node registerUser.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Successfully registered and enrolled admin user "user1" and imported it into the wallet
root@CK:~/prac7/fabric-samples/fabcar/javascript#
```

First, let's run our query.js program to return a listing of all the cars on the ledger. This program uses our second identity – user1 – to access the ledger:

node query.js

```
root@CK:~/prac7/fabric-samples/fabcar/javascript# node query.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is: [{"Key": "CAR0", "Record": {"color": "blue", "docType": "car", "make": "Toyota", "model": "Prius", "owner": "Tomoko"}}, {"Key": "CAR1", "Record": {"color": "red", "docType": "car", "make": "Ford", "model": "Mustang", "owner": "Brad"}}, {"Key": "CAR2", "Record": {"color": "green", "docType": "car", "make": "Hyundai", "model": "Tucson", "owner": "Jin Soo"}}, {"Key": "CAR3", "Record": {"color": "yellow", "docType": "car", "make": "Volkswagen", "model": "Passat", "owner": "Max"}}, {"Key": "CAR4", "Record": {"color": "black", "docType": "car", "make": "Tesla", "model": "S", "owner": "Adriana"}}, {"Key": "CAR5", "Record": {"color": "purple", "docType": "car", "make": "Peugeot", "model": "205", "owner": "Michel"}}, {"Key": "CAR6", "Record": {"color": "white", "docType": "car", "make": "Chery", "model": "S2L", "owner": "Aarav"}}, {"Key": "CAR7", "Record": {"color": "violet", "docType": "car", "make": "Fiat", "model": "Punto", "owner": "Pari"}}, {"Key": "CAR8", "Record": {"color": "indigo", "docType": "car", "make": "Tata", "model": "Nano", "owner": "Valeria"}}, {"Key": "CAR9", "Record": {"color": "brown", "docType": "car", "make": "Holden", "model": "Barina", "owner": "Shotaro"}}]
root@CK:~/prac7/fabric-samples/fabcar/javascript#
```

Here run a command cat query.js command where we can see “queryAllCars”

```
root@CK: ~/prac7/fabric-samples/fabcar/javascript# cat query.js
/*
 * SPDX-License-Identifier: Apache-2.0
 */

'use strict';

const { FileSystemWallet, Gateway } = require('fabric-network');
const path = require('path');

const ccpPath = path.resolve(__dirname, '.', '..', 'first-network', 'connection-org1.json');

async function main() {
    try {

        // Create a new file system based wallet for managing identities.
        const walletPath = path.join(process.cwd(), 'wallet');
        const wallet = new FileSystemWallet(walletPath);
        console.log(`Wallet path: ${walletPath}`);

        // Check to see if we've already enrolled the user.
        const userExists = await wallet.exists('user1');
        if (!userExists) {
            console.log('An identity for the user "user1" does not exist in the wallet');
            console.log('Run the registerUser.js application before retrying');
            return;
        }

        // Create a new gateway for connecting to our peer node.
        const gateway = new Gateway();
        gateway.identity = 'user1';
        gateway.wallet = wallet;
        gateway.connection = await gateway.connect(ccpPath, { timeout: 10 });
        gateway.channel = await gateway.getNetwork('mychannel');

        // Get the contract from the network.
        const fabricContract = await gateway.getContract('CarTradeContract');
        const car = await fabricContract.evaluateTransaction('queryAllCars');

        console.log(`Car: ${car}`);
    } catch (err) {
        console.error(`Failed to query car. Error: ${err}`);
        process.exit(1);
    }
}
```

```

// Get the network (channel) our contract is deployed to.
const network = await gateway.getNetwork('mychannel');

// Get the contract from the network.
const contract = network.getContract('fabcar');

// Evaluate the specified transaction.
// queryCar transaction - requires 1 argument, ex: ('queryCar', 'CAR4')
// queryAllCars transaction - requires no arguments, ex: ('queryAllCars')
const result = await contract.evaluateTransaction('queryCar', 'CAR4');
console.log(`Transaction has been evaluated, result is: ${result.toString()}`);

} catch (error) {
  console.error(`Failed to evaluate transaction: ${error}`);
  process.exit(1);
}

main();
root@CK:~/prac7/fabric-samples/fabcar/javascript#

```

Convert it to the below following by using nano command:

```
const result = await contract.evaluateTransaction('queryCar', 'CAR4');
```

```

// Evaluate the specified transaction.
// queryCar transaction - requires 1 argument, ex: ('queryCar', 'CAR4')
// queryAllCars transaction - requires no arguments, ex: ('queryAllCars')
//const result = await contract.evaluateTransaction('queryAllCars');
const result = await contract.evaluateTransaction('queryCar', 'CAR4');
console.log(`Transaction has been evaluated, result is: ${result.toString()}`);

} catch (error) {
  console.error(`Failed to evaluate transaction: ${error}`);
  process.exit(1);
}

main();

```

Nano Editor Key Bindings:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- M-U Undo
- M-A Set Mark
- ^X Exit
- ^R Read File
- ^V Replace
- ^U Paste
- ^J Justify
- ^L Go To Line
- M-E Redo
- M-6 Copy

```

[1]+  Stopped                  nano query.js
root@CK:~/prac7/fabric-samples/fabcar/javascript# node query.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is: {"color":"black","docType":"car","make":"Tesla","model":"S","owner":"Adriana"}
}
root@CK:~/prac7/fabric-samples/fabcar/javascript#

```

change the argument from CAR4 to CAR12

```

// queryCar transaction - requires 1 argument, ex: ('queryCar', 'CAR4')
// queryAllCars transaction - requires no arguments, ex: ('queryAllCars')
//const result = await contract.evaluateTransaction('queryAllCars');
const result = await contract.evaluateTransaction('queryCar', 'CAR12');
console.log(`Transaction has been evaluated, result is: ${result.toString()}`);

} catch (error) {
  console.error(`Failed to evaluate transaction: ${error}`);
  process.exit(1);
}

main();


```

Nano Editor Key Bindings:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- M-U Undo
- M-A Set Mark
- ^X Exit
- ^R Read File
- ^V Replace
- ^U Paste
- ^J Justify
- ^L Go To Line
- M-E Redo
- M-6 Copy

```
node invoke.js
```

```
[2]+ Stopped nano query.js
root@CK:~/prac7/fabric-samples/fabcar/javascript# node invoke.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Transaction has been submitted
root@CK:~/prac7/fabric-samples/fabcar/javascript#
```

```
enrollAdmin.js invoke.js node_modules package-lock.json package.json query.js registerUser.js wallet
root@CK:~/prac7/fabric-samples/fabcar/javascript# node query.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is: {"color": "Black", "docType": "car", "make": "Honda", "model": "Accord", "owner": "Tom"
"}
root@CK:~/prac7/fabric-samples/fabcar/javascript#
```

Again, go back to invoke.js and change the smart contract transaction from createCar to changeCarOwner with a corresponding change in input arguments:

```
await contract.submitTransaction('changeCarOwner', 'CAR12', 'Dave');
```

The first argument — CAR12 — identifies the car that will be changing owners.

The second argument — Dave — defines the new owner of the car

Save and execute the program again: node invoke.js

```
[10]+ Stopped nano query.js
root@CK:~/prac7/fabric-samples/fabcar/javascript# node query.js
Wallet path: /root/prac7/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is: {"color": "Black", "docType": "car", "make": "Honda", "model": "Accord", "owner": "Dave"
"}
root@CK:~/prac7/fabric-samples/fabcar/javascript# |
```

Practical 8

Aim: Develop a voting application using Hyperledger and Ethereum.

Step: Open Remix IDE (<https://remix.ethereum.org/>)

Create a vote.sol file and enter the code

The screenshot shows the Remix IDE interface. On the left is the FILE EXPLORER panel, which lists workspaces, states, artifacts, contracts, scripts, tests, and various configuration files like prettierc.json and README.txt. A file named 'vote.sol' is selected in the list. The main right-hand area is a code editor containing the following Solidity code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Voting {
    address public admin;
    struct Candidate {
        string name;
        uint voteCount;
    }
    struct Voter {
        bool hasVoted;
        uint votedCandidate;
    }
    mapping(uint => Candidate) public candidates;
    mapping(address => Voter) public voters;
    uint public candidatesCount;
    bool public votingEnded;

    event Voted(address indexed voter, uint indexed candidateId);
    event CandidateAdded(string name, uint candidateId);
    event VotingEnded();

    modifier onlyAdmin() {
        require(msg.sender == admin, "Only admin can perform this action.");
        _;
    }

    modifier onlyBeforeEnd() {
        require(!votingEnded, "Voting has ended.");
        _;
    }

    constructor() {
        admin = msg.sender;
    }
}
```

Now compile the code

The screenshot shows the Remix IDE with the SOLIDITY COMPILER tab active. The compiler version is set to 0.8.26+commit.8a97fa7a. The code editor contains the same Solidity Voting contract code as the previous screenshot. The interface includes various developer tools and configurations on the left side.

Deploy the contract

The screenshot shows the Truffle UI interface. On the left, there's a sidebar with various icons. The main area has tabs for "DEPLOY & RUN TRANSACTIONS" and "CONTRACT". Under "CONTRACT", it says "Voting - vote.sol" with an "evm version: cancan" dropdown. There's a "Deploy" button and a checkbox for "Publish to IPFS". Below that are buttons for "At Address" and "Load contract from Address". A list of "Transactions recorded" and "Deployed Contracts" is shown, with "VOTING AT 0X7B9...B6ACE" expanded. It lists several functions: "addCandidate" (with parameter "string _name"), "endVoting", "vote" (with parameter "uint256 _candidateId"), "admin", "candidates" (with parameter "uint256"), "candidatesCo...", "getWinner", "voters" (with parameter "address"), and "votingEnded". At the bottom, there's a "Low level interactions" section with a "CALLDATA" button and a "Transact" button. The right side shows the Solidity code for the Voting contract.

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract Voting {
5     address public admin;
6     struct Candidate {
7         string name;
8         uint voteCount;
9     }
10    struct Voter {
11        bool hasVoted;
12        uint votedCandidate;
13    }
14
15    mapping(uint => Candidate) public candidates;
16    mapping(address => Voter) public voters;
17    uint public candidatesCount;
18    bool public votingEnded;
19
20    event Voted(address indexed voter, uint indexed candidateId);
21    event CandidateAdded(string name, uint candidateId);
22    event VotingEnded();
23
24    modifier onlyAdmin() {
25        require(msg.sender == admin, "Only admin can perform this action.");
26       _;
27    }
28
29    modifier onlyBeforeEnd() {
30        require(!votingEnded, "Voting has ended.");
31       _;
32    }
33
34    constructor() {
35        admin = msg.sender;
36    }

```

[vm] from: 0x5B3...addC4 to: Voting.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x3ba...a96a2

Here you can add a candidate vote see the voters candidate and even see the result after ending

Here you can add candidates and see the count

This screenshot is similar to the one above but focuses on the "candidates" function. The "candidates" field in the UI shows "0: uint256: 3". The right side shows the same Solidity code as before, with the addition of a call to the "candidatesCount" function.

```

14
15    mapping(uint => Candidate) public candidates;
16    mapping(address => Voter) public voters;
17    uint public candidatesCount;
18    bool public votingEnded;
19
20    event Voted(address indexed voter, uint indexed candidateId);
21    event CandidateAdded(string name, uint candidateId);
22    event VotingEnded();
23
24    modifier onlyAdmin() {
25        require(msg.sender == admin, "Only admin can perform this action.");
26       _;
27    }
28
29    modifier onlyBeforeEnd() {
30        require(!votingEnded, "Voting has ended.");
31       _;
32    }
33
34    constructor() {
35        admin = msg.sender;
36    }

```

call to Voting.candidatesCount

call [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Voting.candidatesCount() data: 0x2d3...5a8a2

On clicking the candidate you can see the details of that candidate



```
mapping(uint => Candidate) public candidates;
mapping(address => Voter) public voters;
uint public candidatesCount;
bool public votingEnded;

event Voted(address indexed voter, uint indexed candidateId);
event CandidateAdded(string name, uint candidateId);
event VotingEnded();

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action.");
    _;
}

modifier onlyBeforeEnd() {
    require(!votingEnded, "Voting has ended.");
    _;
}

constructor() { 956431 gas 931200 gas
    admin = msg.sender;
}

call [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Voting.candidates(uint256) data: 0x347...00001
```

Now change the account and vote using the id and see end the voting to see the results.



```
uint voteCount;
}

struct Voter {
    bool hasVoted;
    uint votedCandidate;
}

mapping(uint => Candidate) public candidates;
mapping(address => Voter) public voters;
uint public candidatesCount;
bool public votingEnded;

event Voted(address indexed voter, uint indexed candidateId);
event CandidateAdded(string name, uint candidateId);
event VotingEnded();

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action.");
    _;
}

modifier onlyBeforeEnd() {
    require(!votingEnded, "Voting has ended.");
    _;
}

constructor() { 956431 gas 931200 gas
    admin = msg.sender;
}

call [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Voting.votingEnded() data: 0x9d5...a0529
```