Write-up	Correctness of Program	Documentation o Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

Group C
Assignment No : 1

Title of the Assignment: Installation of MetaMask and study spending Ether per transaction

Objective of the Assignment: Students should be able to learn new technology such as metamask.Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

Contents for Theory:

1. **Introduction Blockchain**
2. **Cryptocurrency**
3. **Transaction Wallets**
4. **Ether transaction**
5. **Installation Process of Metamask**

Introduction to Blockchain

- Blockchain can be described as a data structure that holds transactional records and while ensuring security, transparency, and decentralization. You can also think of it as a chain or records stored in the forms of blocks which are controlled by no single authority.
- A blockchain is a distributed ledger that is completely open to any and everyone on the network. Once an information is stored on a blockchain, it is extremely difficult to change or alter it.
- Each transaction on a blockchain is secured with a digital signature that proves its authenticity. Due to the use of encryption and digital signatures, the data stored on the blockchain is tamper-proof and cannot be changed.
- Blockchain technology allows all the network participants to reach an agreement, commonly known as consensus. All the data stored on a blockchain is recorded digitally and has a common history which is available for all the network participants. This way, the chances of any fraudulent activity or duplication of transactions is eliminated without the need of a third-party.

Blockchain Features

The following features make the revolutionary technology of blockchain stand out:

- **Decentralized**

  Blockchains are decentralized in nature meaning that no single person or group holds the authority of the overall network. While everybody in the network has the copy of the distributed ledger with them, no one can modify it on his or her own. This unique feature of blockchain allows transparency and security while giving power to the users.

- **Peer-to-Peer Network**

  With the use of Blockchain, the interaction between two parties through a peer-to-peer model is easily accomplished without the requirement of any third party. Blockchain uses P2P protocol which allows all the network participants to hold an identical copy of transactions, enabling approval through a machine consensus. For example, if you wish to make any transaction from one part of the world to another, you can do that with blockchain all by yourself within a few seconds. Moreover, any interruptions or extra charges will not be deducted in the transfer.

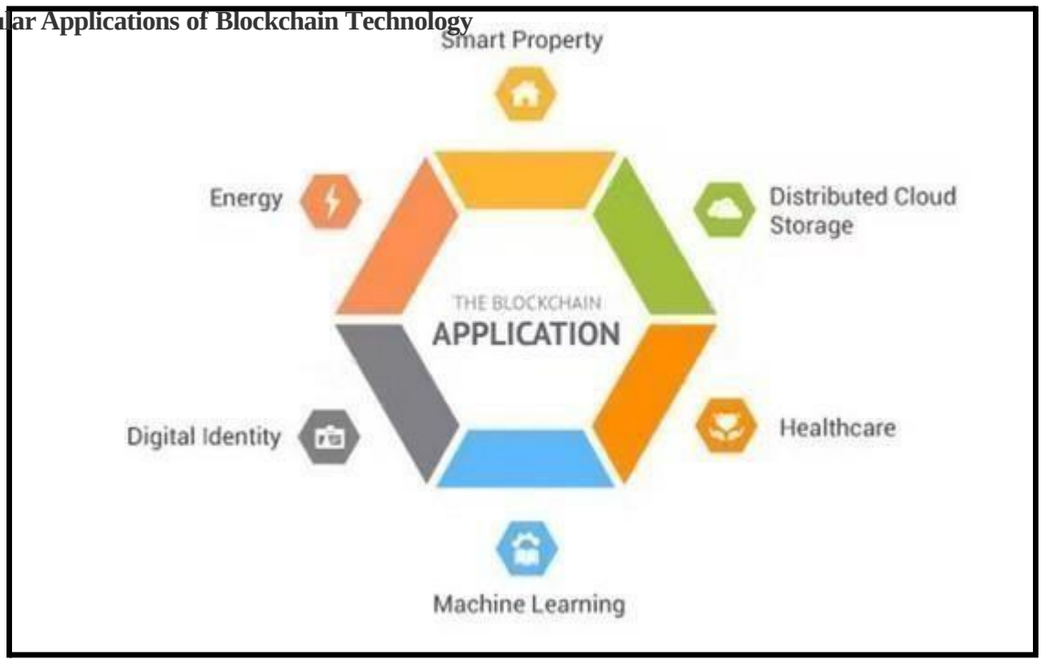
- **Immutable**

  The immutability property of a blockchain refers to the fact that any data once written on the blockchain cannot be changed. To understand immutability, consider sending email as an example. Once you send an email to a bunch of people, you cannot take it back. In order to find a way around, you'll have to ask all the recipients to delete your email which is pretty tedious. This is how immutability works.

- **Tamper-Proof**

  With the property of immutability embedded in blockchains, it becomes easier to detect tampering of any data. Blockchains are considered tamper-proof as any change in even one single block can be detected and addressed smoothly. There are two key ways of detecting tampering namely, hashes and blocks.

Popular Applications of Blockchain Technology



Benefits of Blockchain Technology:

- **Time-saving:** No central Authority verification needed for settlements making the process faster and cheaper.
- **Cost-saving:** A Blockchain network reduces expenses in several ways. No need for third-party verification. Participants can share assets directly. Intermediaries are reduced. Transaction efforts are minimized as every participant has a copy of shared ledger.
- **Tighter security:** No one can temper with Blockchain Data as it is shared among

millions of participants. The system is safe against cybercrimes and Fraud.

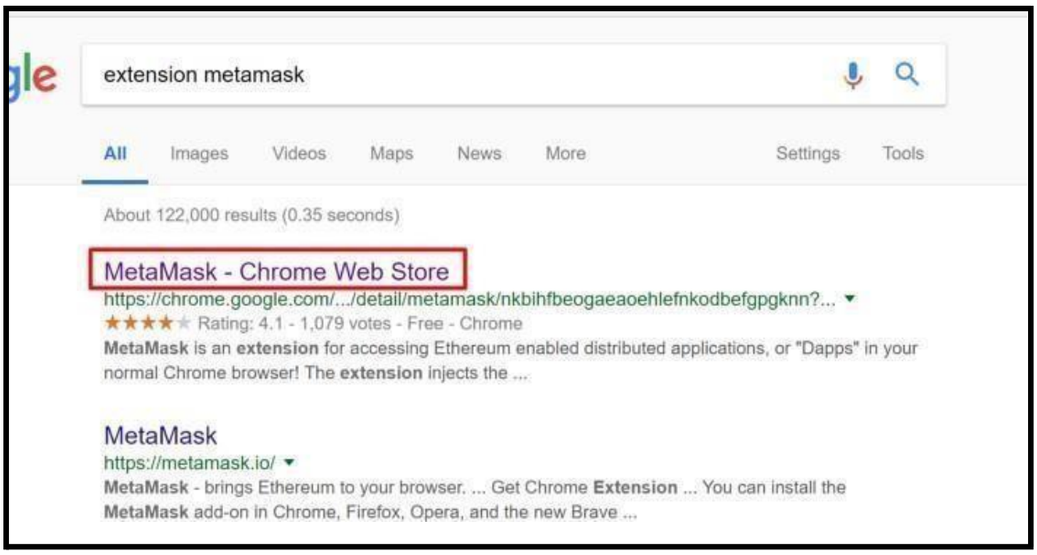
- In finance market trading, Fibonacci retracement levels are widely used in technical analysis.

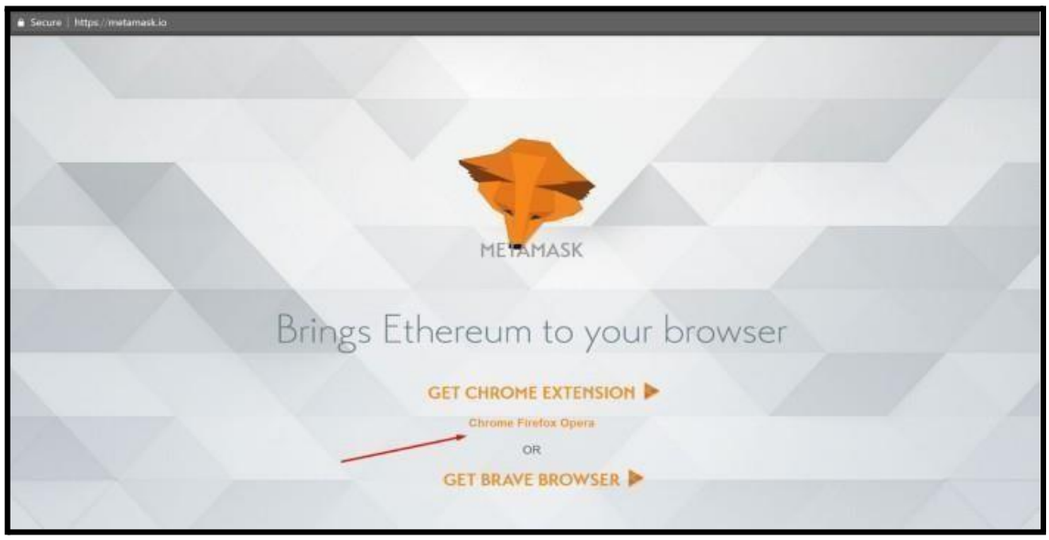
How to use MetaMask: A step by step guide

MetaMask is one of the most popular browser extensions that serves as a way of storing your Ethereum and other ERC-20 Tokens. The extension is free and secure, allowing web applications to read and interact with Ethereum‘s blockchain.

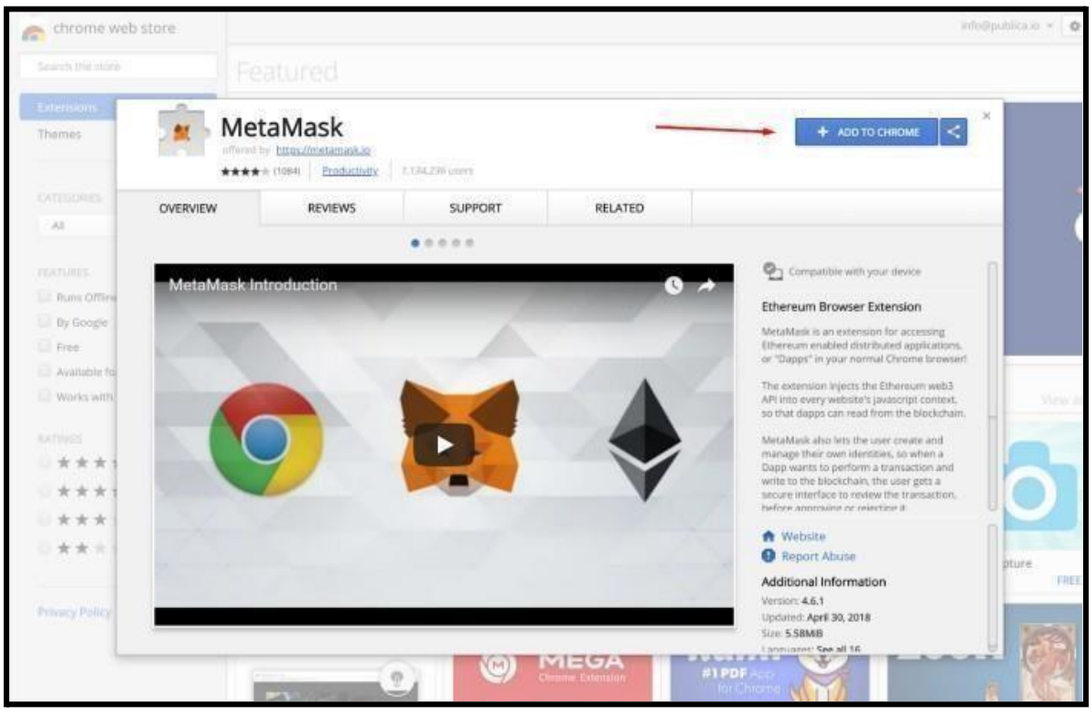
Step 1. Install MetaMask on your browser.

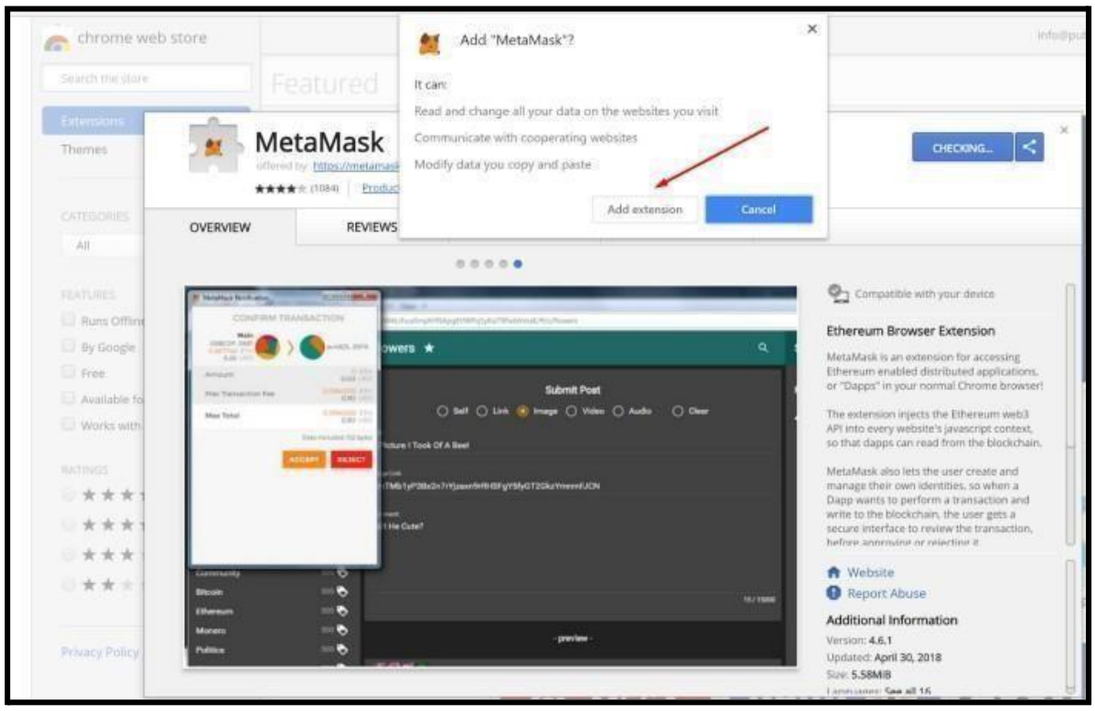
To create a new wallet, you have to install the extension first. Depending on your browser, there are different marketplaces to find it. Most browsers have MetaMask on their stores, so it‘s not that hard to see it, but either way, here they are Chrome, Firefox, and Opera.





- Click on **Install MetaMask** as a Google Chrome extension.
- Click **Add to Chrome**.
- Click **Add Extension**.

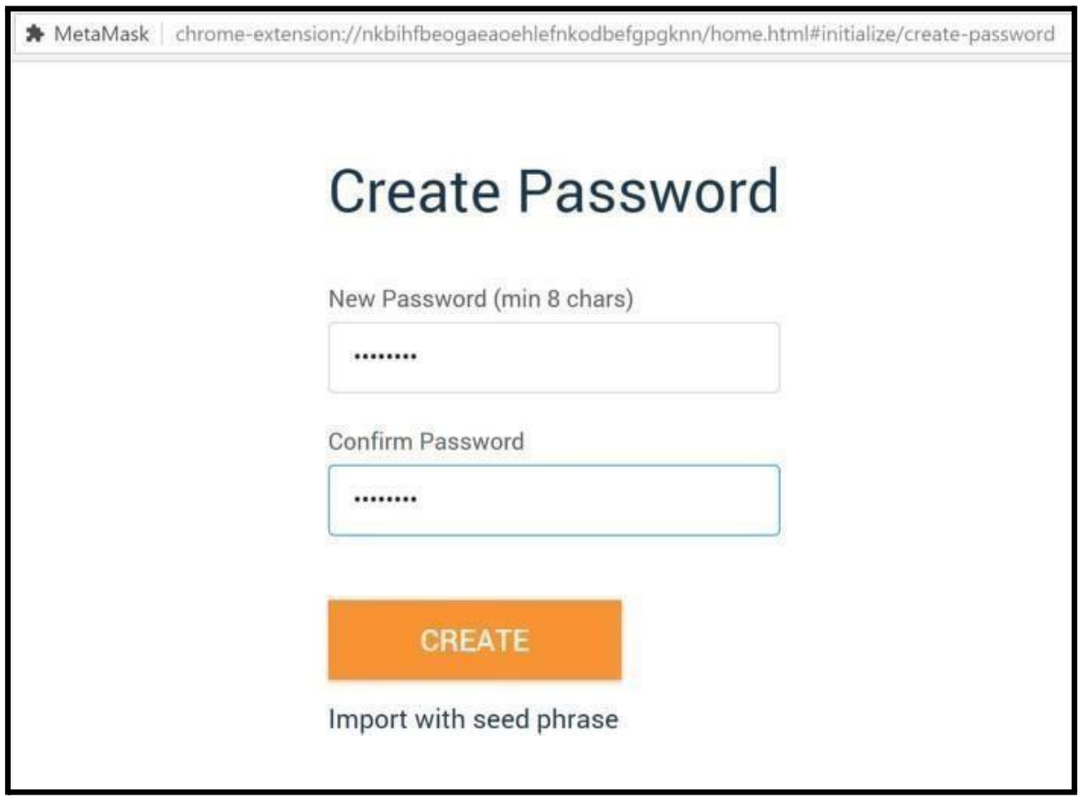




nd it‘s as easy as that to install the extension on your browser, continue reading the next step to figure out how to create an account.

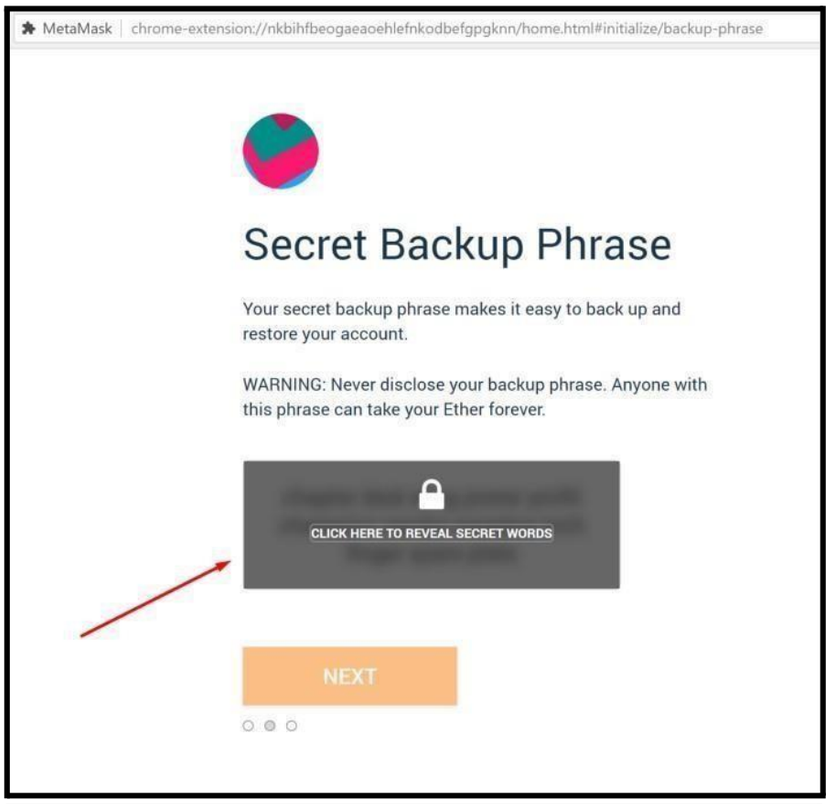
Step 2. Create an account.

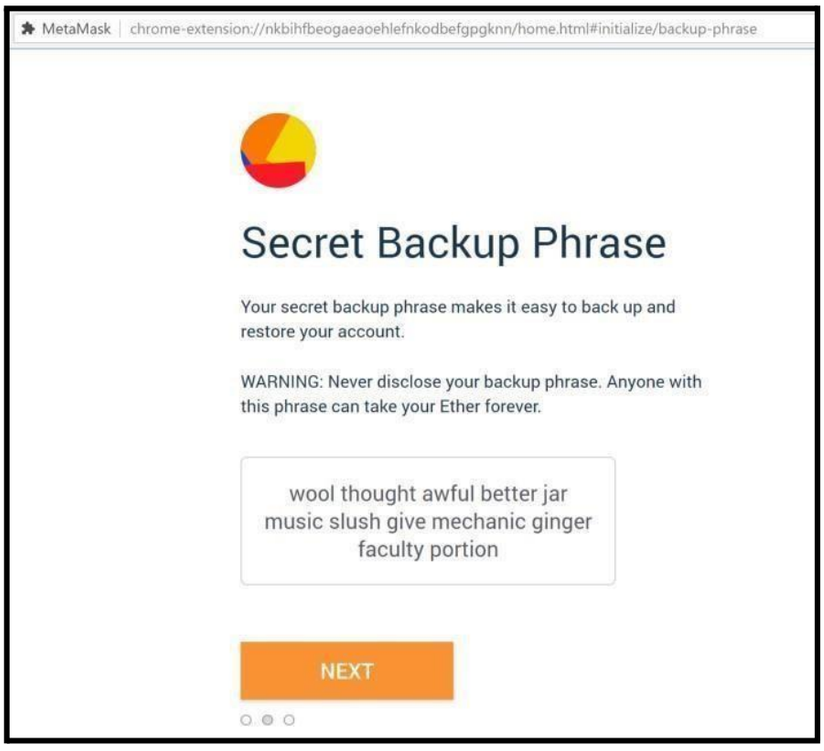
- Click on the extension icon in the upper right corner to open MetaMask.
- To install the latest version and be up to date, **click Try it now**.
- **Click Continue**.
- You will be prompted to create a new password. **Click Create**.



- Proceed by **clicking Next** and accept the Terms of Use.

Click Reveal Secret Words. There you will see a 12 words seed phrase. This is really important and usually not a good idea to store digitally, so take your time and write it down





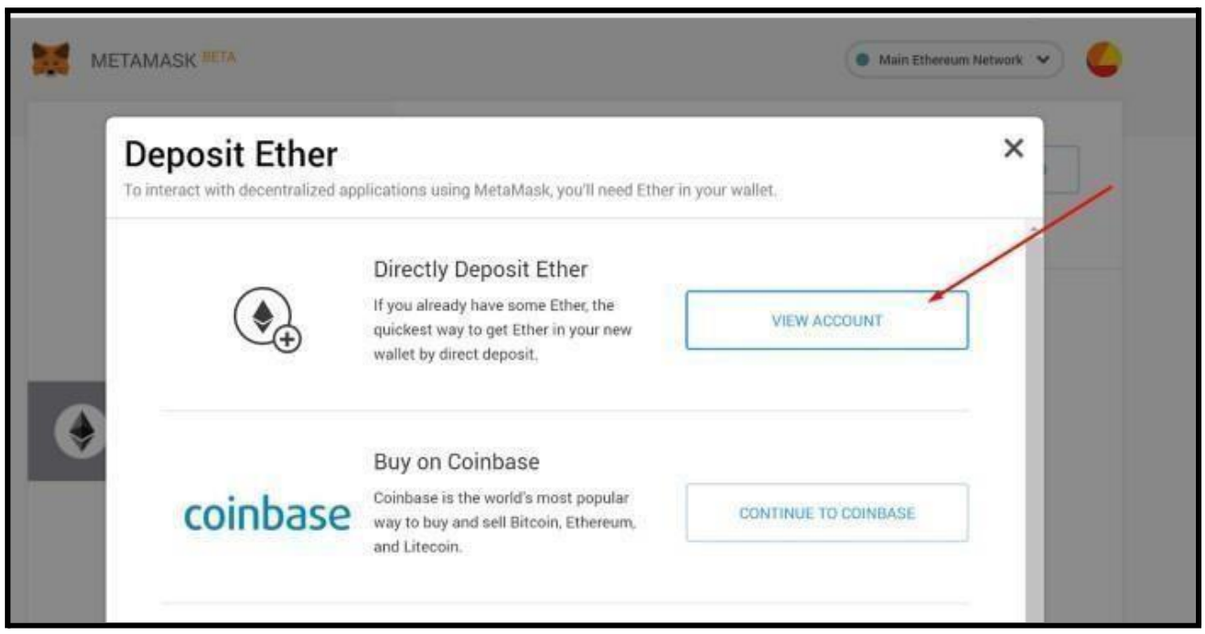
- Verify your secret phrase by selecting the previously generated phrase in order. **Click Confirm**.

And that‘s it; now you have created your MetaMask account successfully. A new Ethereum wallet

address has just been created for you. It‘s waiting for you to deposit funds, and if you want to learn how to do that, look at the next step below.

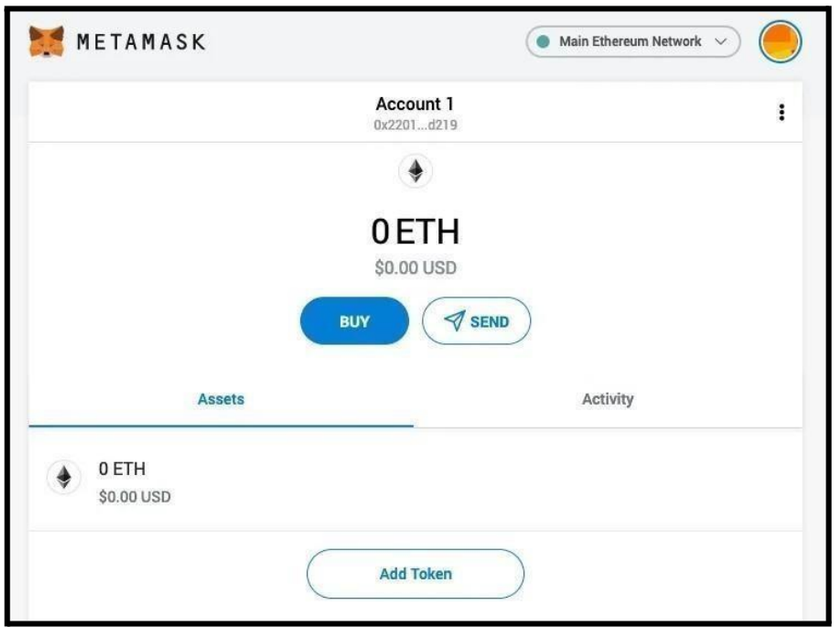
Step 3. Depositing funds.

- Click on **View Account**.



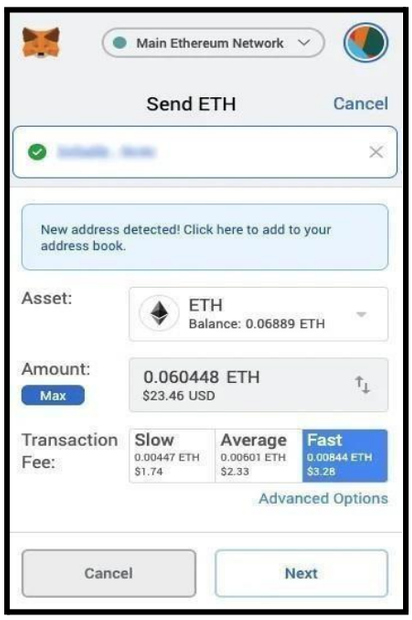
You can now see your public address and share it with other people. There are some methods to buy coins offered by MetaMask, but you can do it differently as well; you just need your address.

If you ever get logged out, you‘ll be able to log back in again by clicking the MetaMask icon, which will have been added to your web browser (usually found next to the URL bar).



You can now access your list of assets in the _Assets‘ tab and view your transaction history in the _Activity‘ tab.

- Sending crypto is as simple as clicking the _Send‘ button, entering the recipient address and amount to send, and selecting a transaction fee. You can also manually adjust the transaction fee using the _Advanced Options‘ button, using information from ETH Gas Station or similar platforms to choose a more acceptable gas price.
- After clicking _Next‘, you will then be able to either confirm or reject the transaction on the subsequent page.



- To use MetaMask to interact with a dapp or smart contract, you'll usually need to find a _Connect to Wallet' button or similar element on the platform you are trying to use. After clicking this, you should then see a prompt asking whether you want to let the dapp connect to your wallet.

What advantages does MetaMask have?

- **Popular** - It is commonly used, so users only need one plugin to access a wide range of dapps.
- **Simple** - Instead of managing private keys, users just need to remember a list of words, and transactions are signed on their behalf.
- **Saves space** - Users don't have to download the Ethereum blockchain, as MetaMask sends requests to nodes outside of the user's computer.
- **Integrated** - Dapps are designed to work with MetaMask, so it becomes much easier to send Ether in and out.

Conclusion- In this way we have explored Concept Blockchain and metamat wallet for transaction of digital currency

Assignment Question

1. **What Are the Different Types of Blockchain Technology?**
2. **What Are the Key Features/Properties of Blockchain?**
3. **What Type of Records You Can Keep in A Blockchain?**
4. **What is the difference between Ethereum and Bitcoin?**
5. **What are Merkle Trees? Explain their concept.**
6. **What is Double Spending in transaction operation**
7. **Give real-life use cases of blockchain.**

Reference link

- https://hackernoon.com/blockchain-technology-explained-introduction-meaning-and-applications-edb d6759a2b2
- https://levelup.gitconnected.com/how-to-use-metamask-a-step-by-step-guide-f380a3943fb1
- https://decrypt.co/resources/metamask

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

0

Group C
Assignment No:2

Title: Create your own wallet using MetaMask for crypto transactions.

Objective: Understand and explore the working of Blockchain technology and its applications.

Course Outcome:

CO6:Interpret the basic concepts in Blockchain technology and its application.

Description:

MetaMask is a cryptocurrency wallet used to interact with the Ethereum Blockchain. It can be accessed through an app or through a browser extension.MetaMask is a popular cryptocurrency wallet known for its ease of use, availability on both desktops and mobile devices, the ability to buy, send, and receive cryptocurrency from within the wallet, and collect non-fungible tokens (NFTs) across two block chains. While experienced crypto users will appreciate the simplicity and fast transactions, those new in the space are at a higher risk of losing their tokens from lost secret phrases, malicious websites, and other cryptocurrency scams

Similarly, you‟ll need a crypto wallet to transact with a Blockchain.

A wallet is your personal key to interact with the cryptographic world. It powers you to buy, sell or transfer assets on the Blockchain.

And MetaMask is a wallet for the most diverse Blockchain in existence–Ethereum. It‟s your gateway to its DeFi ecosystem, non-fungible tokens (NFTs), ERC-20 tokens, and practically–everything Ethereum.

It‟s available as an app for iOS and Android. In addition, you can use this as an extension with a few web browsers: Chrome, Firefox, Brave, and Edge.

Let‟s take a look at some of the prominent features of this wallet:

Ease of use

Starting with MetaMask is easy, quick, and anonymous. You don‟t even need an email address. Just set up a password and remember (and store) the secret recovery phrase, and you‟re done.

Security

Your information is encrypted in your browser that nobody has access to. In the event of a lost password, you have the 12-word secret recovery phase (also called a seed phrase) for

recovery. Notably, it‟s essential to keep the seed phrase safe, as even MetaMask has no information about it. Once lost, it can‟t be retrieved.

Built-In Crypto Store

If you‟re wondering, no, you can‟t buy Bitcoin with MetaMask. It only supports Ether and other Ether-related tokens, including the famous ERC-20 tokens. Cryptocurrencies (excluding Ether) on Ethereum are built as ERC-20 tokens.

Backup and Restore

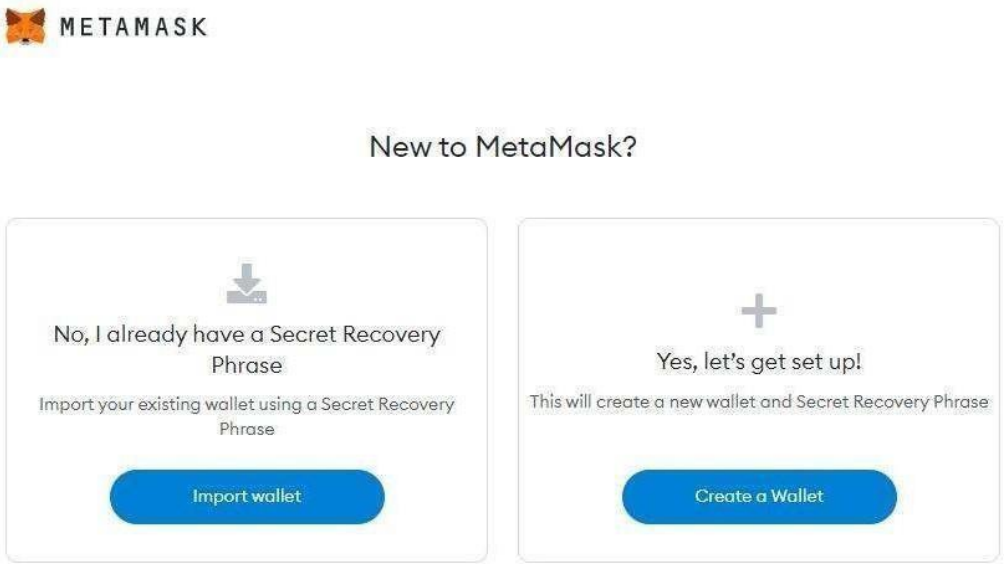
MetaMask stores your information locally. So, in case you switch browsers or machines, you can restore your MetaMask wallet with your secret recovery phrase.

Community Support

As of August 2021, MetaMask was home to 10 million monthly active users around the world. It‟s simple and intuitive user interface keeps pushing these numbers with a recorded 1800% increase from July 2020.

Conclusively, try MetaMask if hot wallets are your pick. Let‟s begin with the installation before moving to its use cases. Further sections entail the illustration for Chrome web browser and Android mobile platform.

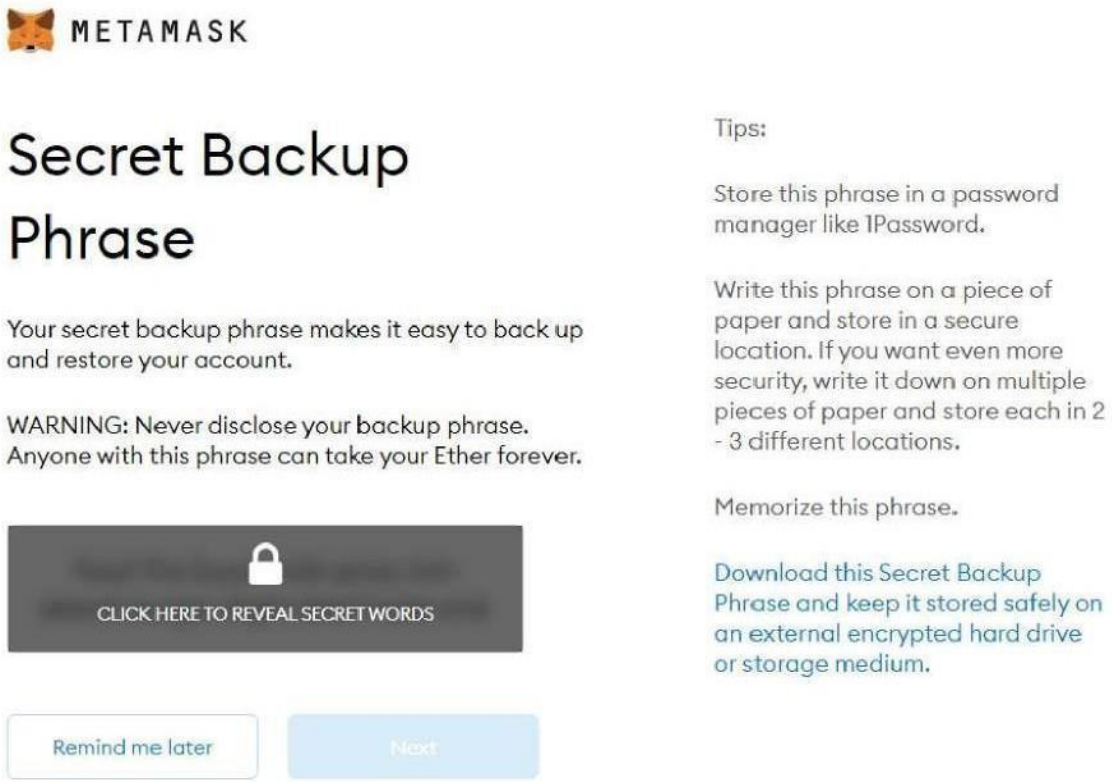
Step 1: Click on the ―Create a wallet‖.



Step 2: Create a password for your wallet. This password is to be entered every time the browser is launched and wants to use MetaMask. A new password needs to be created if chrome is uninstalled or if there is a switching of browsers. In that case, go through the Import Wallet button. This is because MetaMask stores the keys in the browser. Agree to Terms of Use.

Step 3:Click on the dark area which says *Click here to reveal secret words* to get your secret phrase.

Step 4: This is the most important step. Back up your secret phrase properly. Do not store your secret phrase on your computer. Please read everything on this screen until you understand it completely before proceeding. The secret phrase is the only way to access your wallet if you forget your password. Once done click the Next button.

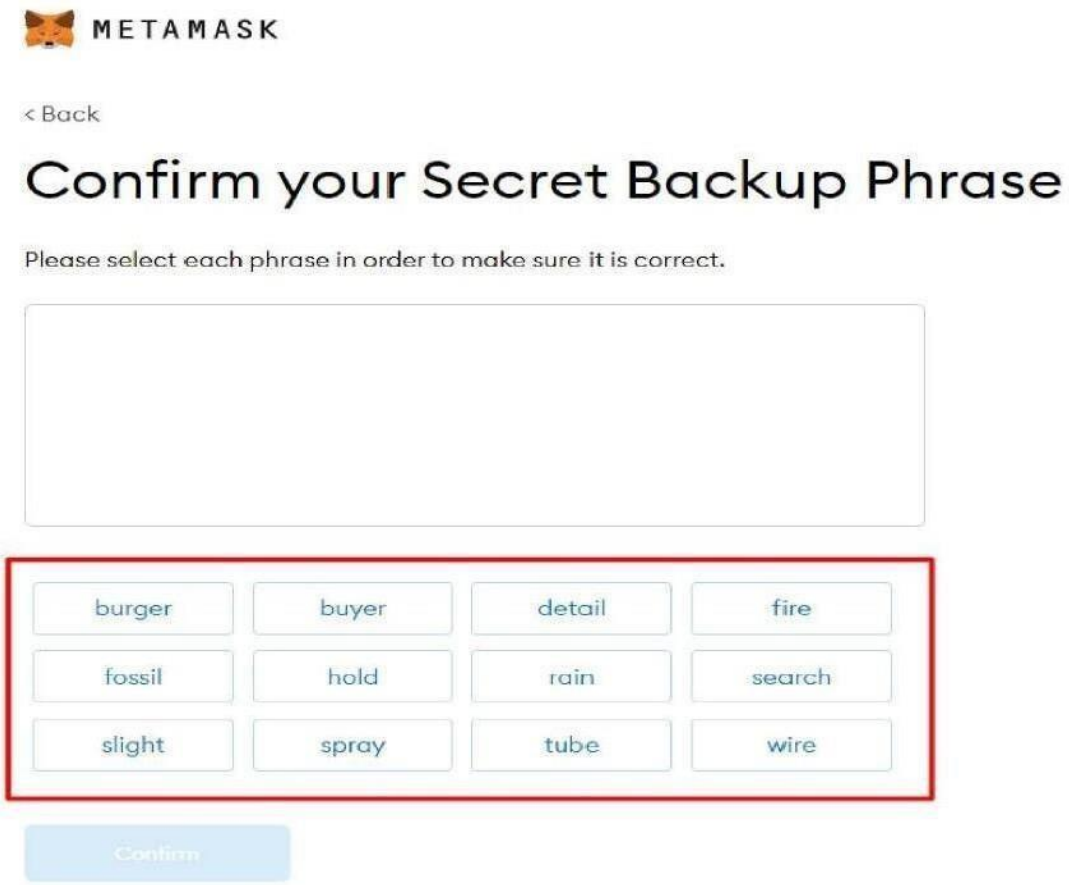


Step 5:

Securely store the seed phrase for your wallet Click on ―Click here to reveal secret words‖ to show the seed phrase.

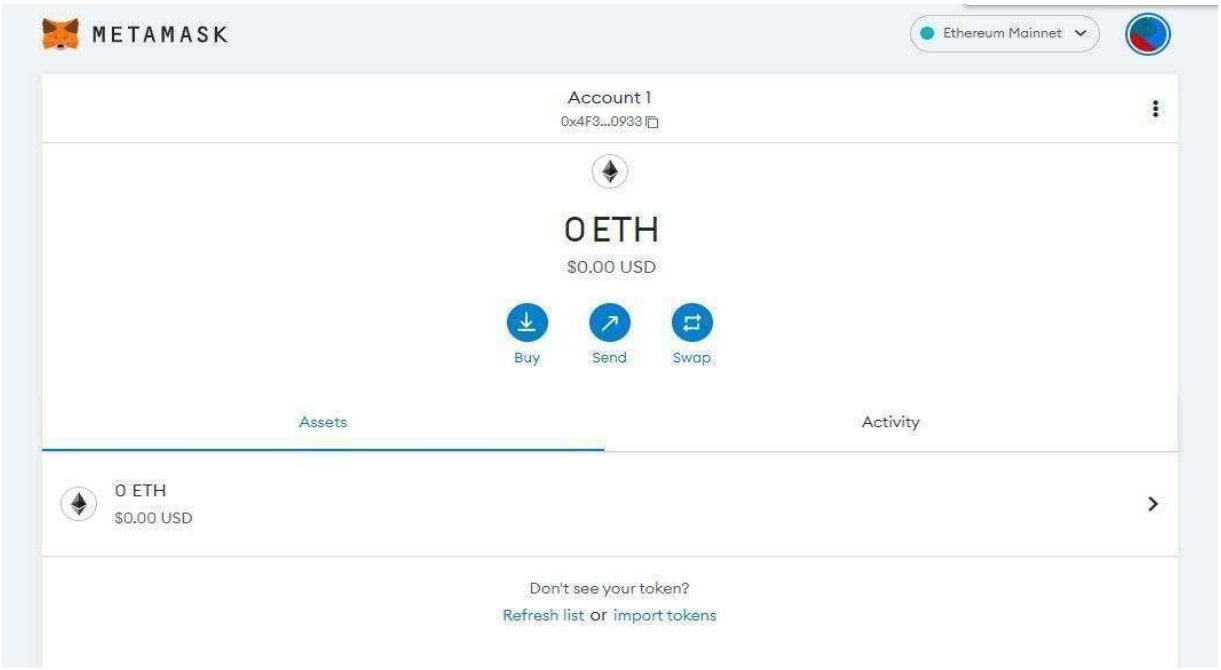
MetaMask requires that you store your seed phrase in a safe place. It is the only way to recover your funds should your device crash or your browser reset. We recommend you write it down. The most common method is to write your 12-word phrase on a piece of paper and store it safely in a place where only you have access. Note: if you lose your seed phrase, MetaMask can‟t help you recover your wallet and your funds will be lost forever.

Never share your seed phrase or your private key to anyone or any site, unless you want them to have full control over your funds.



Step 6: Click the *Confirm* button. Please follow the tips mentioned.

Step 7: One can see the balance and copy the address of the account by clicking on the *Account 1* area.



Conclusion: Hence,wehavestudiedto create a wallet using Meta

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

—----------

Assignment No. 3

Title: Write a smart contract on a test network for Bank account of a customer for following operations.

- Deposit Money
- Withdraw Money
- Show Balance

Objective: Understand and explore the working of Blockchain technology and its applications.

Course Outcome:

CO6:Interpret the basic concepts in Blockchain technology and its application.

Description:

First of all, we need to understand the differences between a paper contract and a smart contract and the reason why smart contracts become increasingly popular and important in recent years. A contract, by definition, is a written or spoken (mostly written) law-enforced agreement containing the rights and duties of the parties. Because most of business contracts are complicated and tricky, the parties need to hire professional agents or lawyers for protecting their own rights. However, if we hire those professionals every time we sign contracts, it is going to be extremely costly and inefficient. Smart contracts perfectly solve this by working on „If-Then‟ principle and also as escrow services. All participants need to put their money, ownership right or other tradable assets into smart contracts before any successful transaction. As long as all participating parties meet the requirement, smart contracts will simultaneously distribute stored assets to recipients and the distribution process will be witnessed and verified by the nodes on Ethereum network.

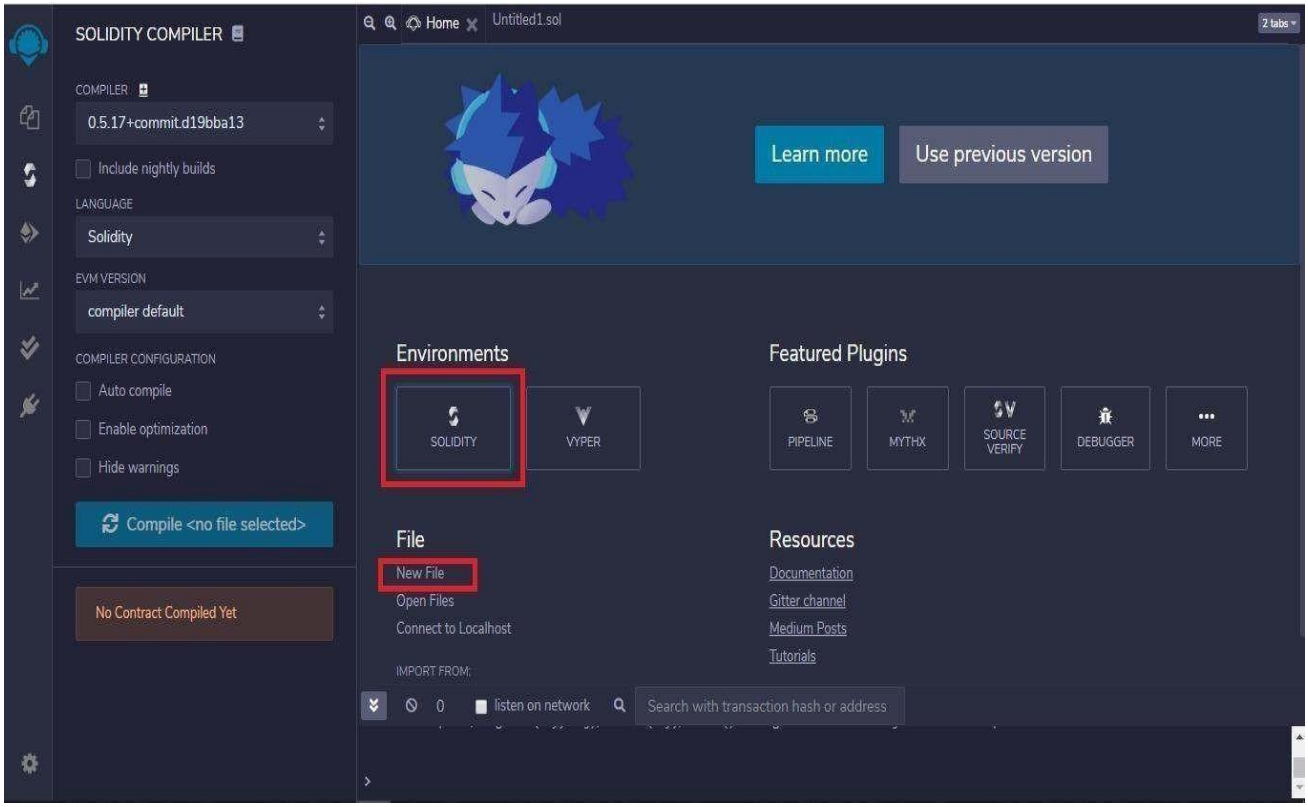
There are a couple of languages we can use to program smart contract. Solidity, an object- oriented and high-level language, is by far the most famous and well maintained one. We can use Solidity to create various smart contracts which can be used in different scenarios, including voting, blind auctions and safe remote purchase. In this lab, we will discuss the semantics and syntax of Solidity with specific explanation, examples and practices

After deciding the coding language, we need to pick an appropriate compiler. Among various compilers like Visual Code Studio, we will use Remix IDE in this and following labs because it can be directly accessed from browser where we can test, debug and deploy smart contracts without any installation.

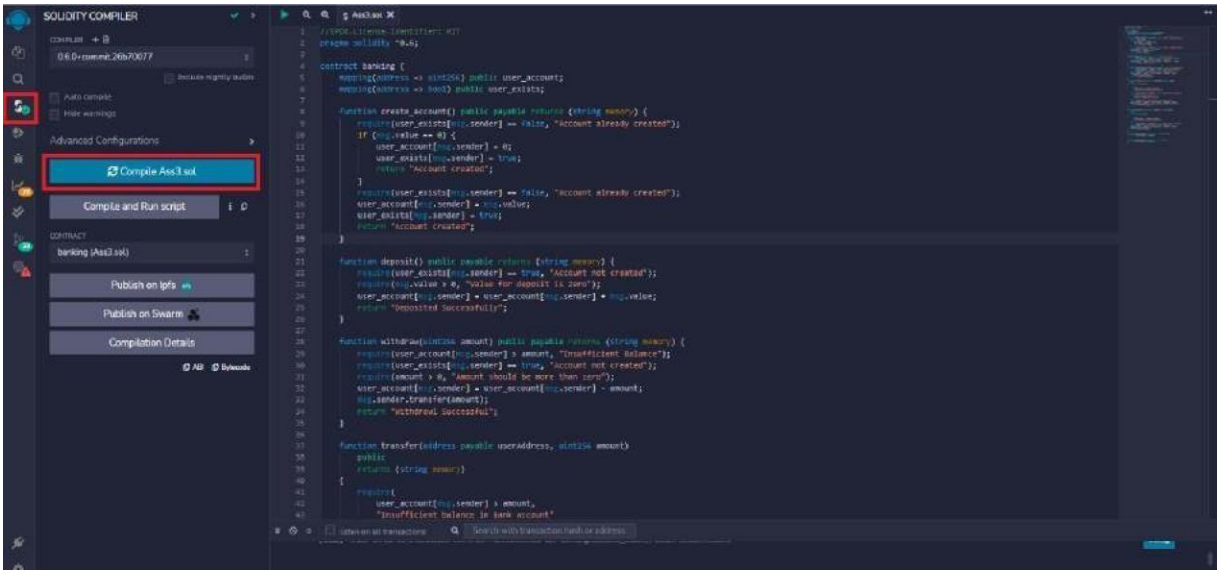
Steps to Execute Solidity Smart Contract using Remix IDE

Remix IDE is generally used to compile and run Solidity smart contracts. Below are the steps for the compilation, execution, and debugging of the smart contract.

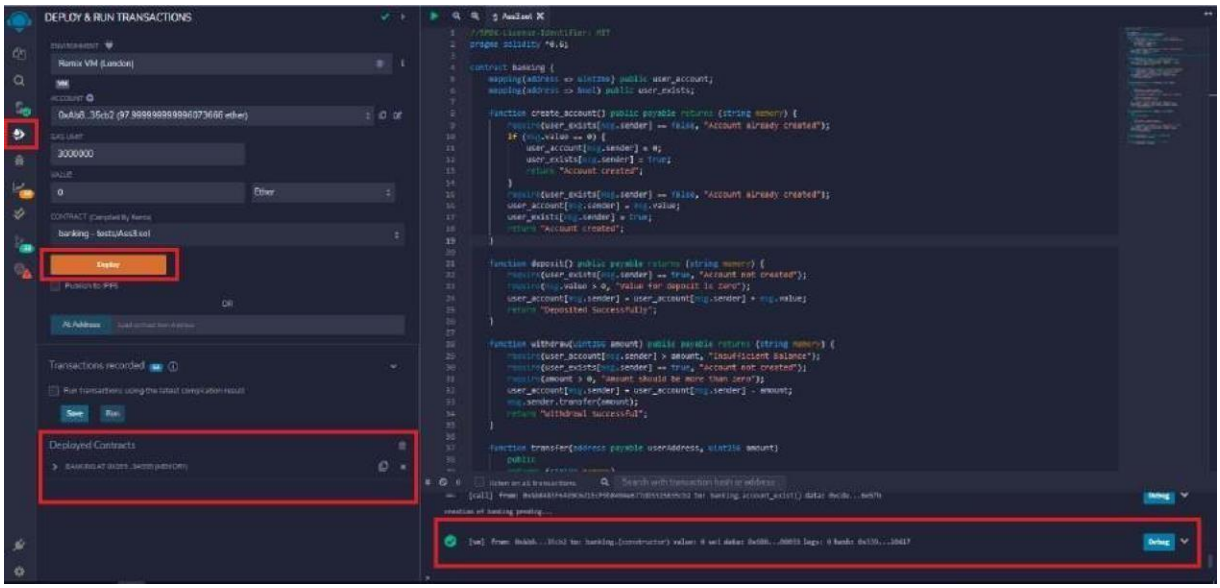
Step 1: Open Remix IDE on any of your browsers, select on the *New File* and click on *Solidity* to choose the environment.



Step 2: Write the Smart contract in the code section, and click the *Compile button* under the Compiler window to compile the contract.



Step 3: To execute the code, click on the *Deploy button* under Deploy and Run Transactions window. After deploying the code click on the drop-down on the console.



Code

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.6;
```

```
contract banking
{
   mapping(address=>uint) public user_account;
   mapping(address=>bool) public user_exists;

   function create_account() public payable returns(string memory)
   {
      require(user_exists[msg.sender]==false,'Account already
      created'); if(msg.value==0)
      {
user_account[msg.sender]=0;
user_exists[msg.sender]=true;
         return "Account created";
      }
      require(user_exists[msg.sender]==false,"Account already created");
user_account[msg.sender]=msg.value;
user_exists[msg.sender]=true;
      return "Account created";
   }

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

   function withdraw(uint amount) public payable returns(string memory)
   {
```

```
    require(user_account[msg.sender]>amount,"Insufficient Balance");
    require(user_exists[msg.sender]==true,"Account not created");
    require(amount>0,"Amount should be more than zero");
    user_account[msg.sender]=user_account[msg.sender]-amount;
    msg.sender.transfer(amount);
        return "Withdrawl Successful";
    }

    function transfer(address payable userAddress, uint amount) public returns(string memory)
    {
        require(user_account[msg.sender]>amount,"Insufficient balance in Bank account");
        require(user_exists[msg.sender]==true,"Account is not created");
        require(user_exists[userAddress]==true,"Transfer account does not exist");
        require(amount>0,"Amount should be more than zero");
user_account[msg.sender]=user_account[msg.sender]-amount;
user_account[userAddress]=user_account[userAddress]+amount;
        return "Transfer Successful";
    }

    function send_amt(address payable toAddress, uint256 amount) public payable
returns(string memory)
    {
        require(user_account[msg.sender]>amount,"Insufficeint balance in Bank account");
        require(user_exists[msg.sender]==true,"Account is not created");
        require(amount>0,"Amount should be more than zero");
user_account[msg.sender]=user_account[msg.sender]-amount;
toAddress.transfer(amount);
        return "Transfer Success";
    }

    function user_balance() public view returns(uint)
```

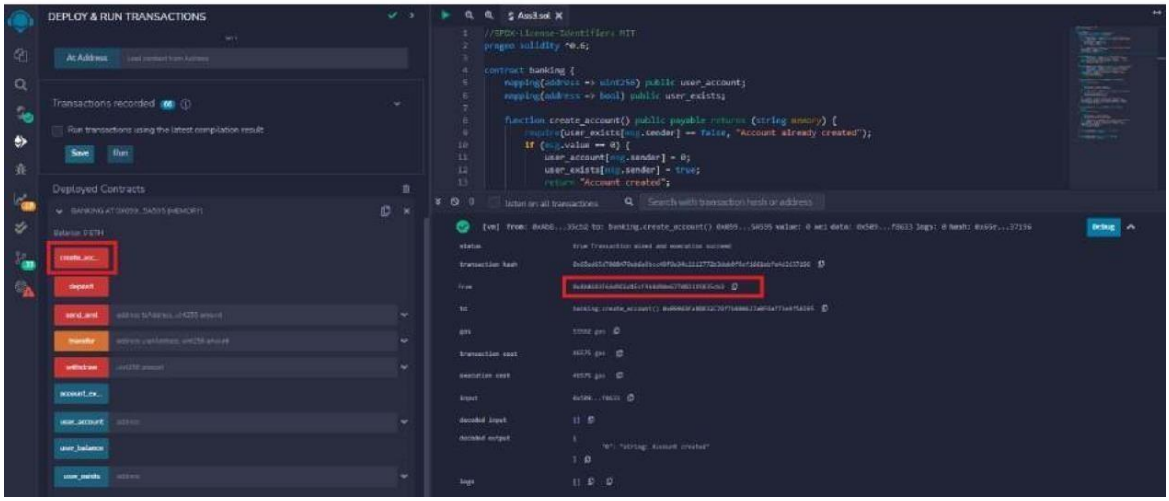
```
    {
        return user_account[msg.sender];
    }

    function account_exist() public view returns(bool)
    {
        return user_exists[msg.sender];
    }
}
```

Sample Output

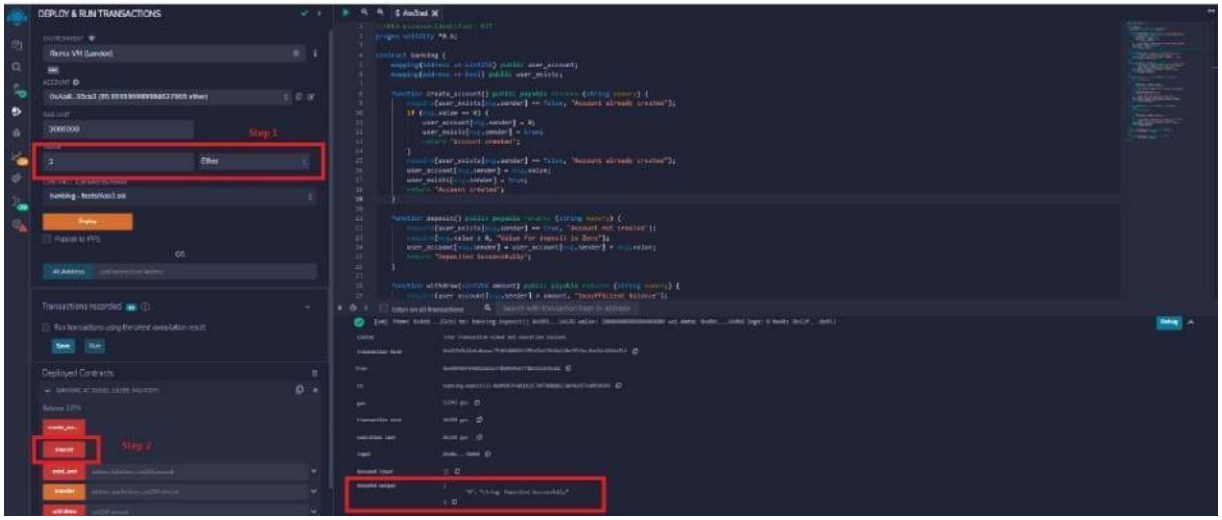
After deploying the contact successful you can observe following buttonscreate_account, deposit, send_amt, transfer, account_exist, user_account, user_balance and user_exists.

Refer the following output

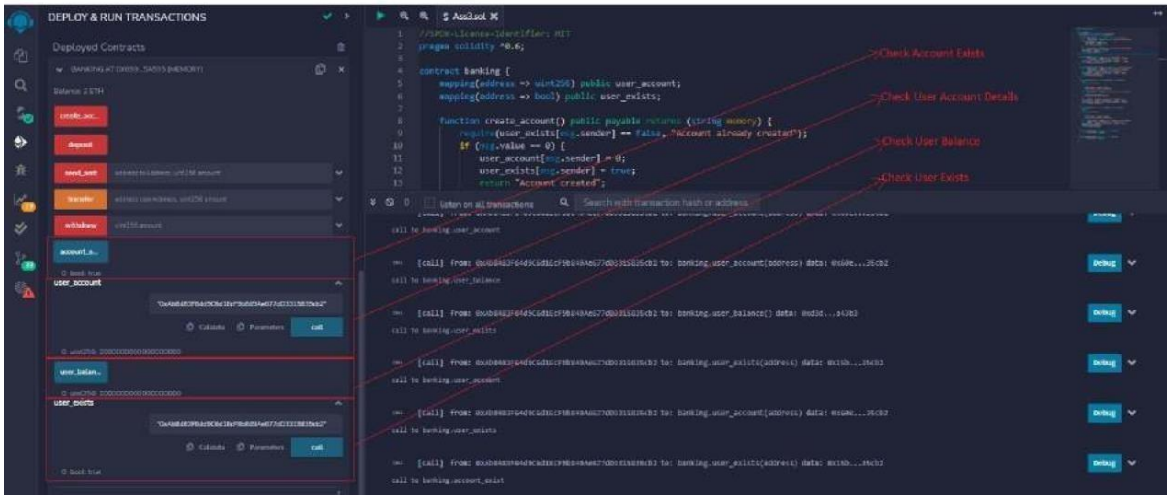
- Create account



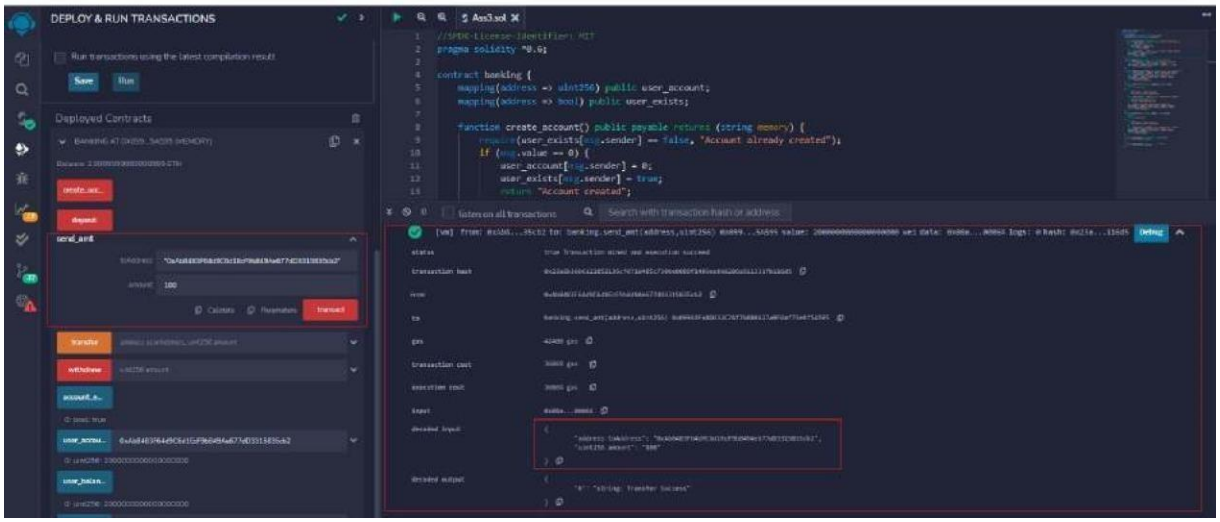
- Deposit Amount



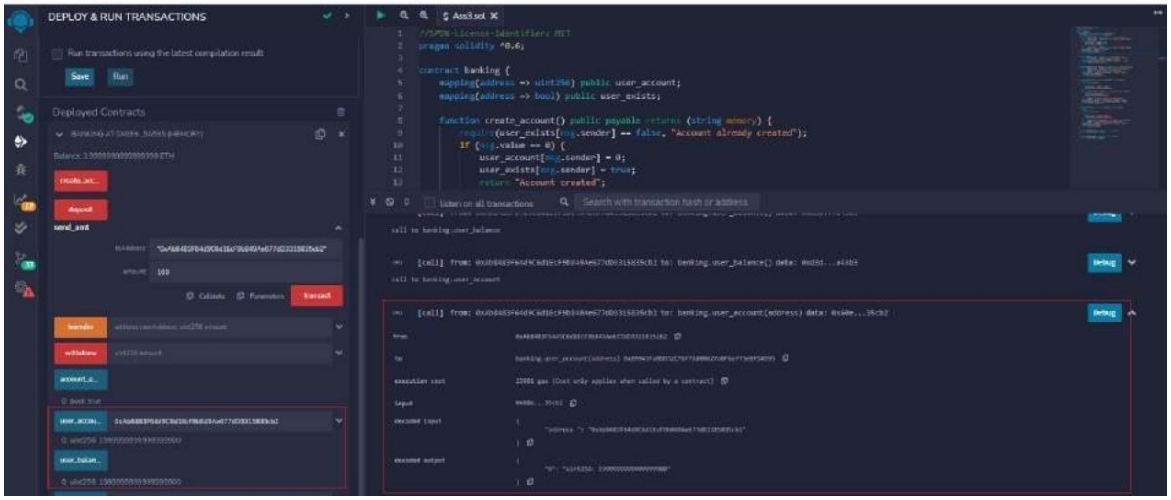
- Check Account Exists
- Check User Account Exists
- Check User Balance
- Check User Exists



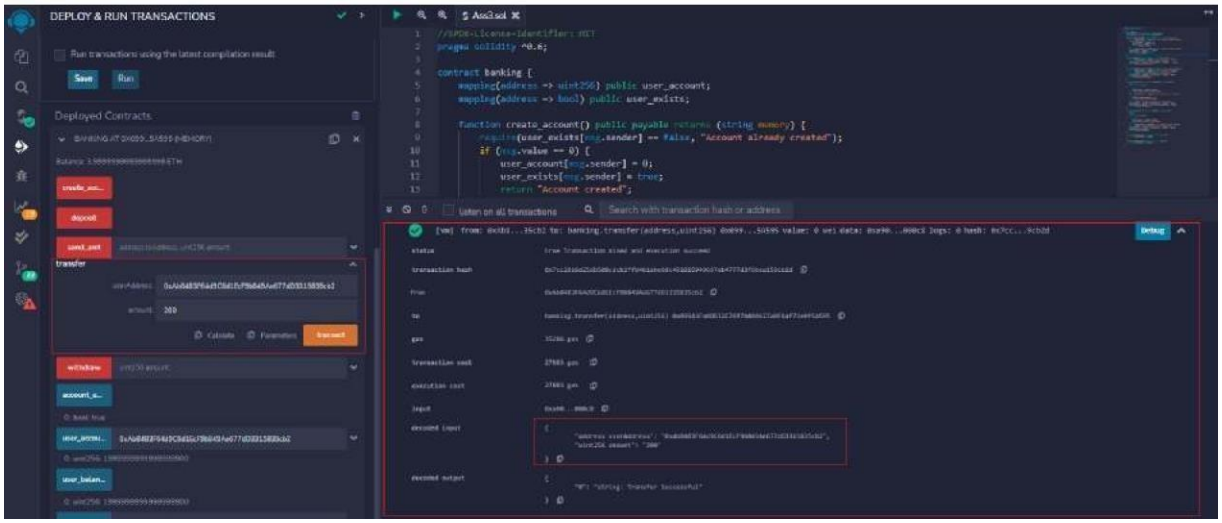
- Send Amount



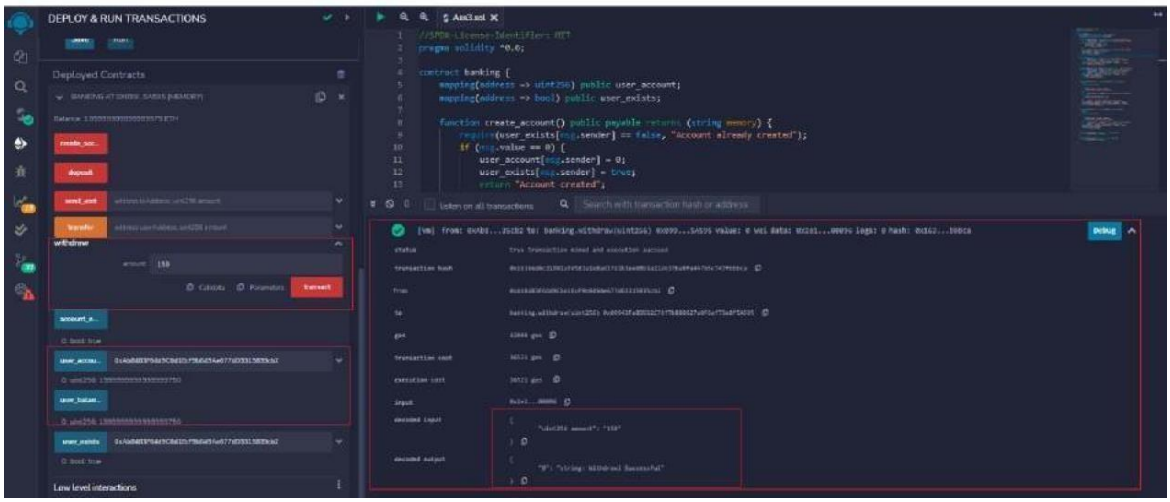
- Check User Account Balance



- Transfer Amount and Check User Account Balance



- Withdraw Amount and Check User Account Balance



Conclusion: Hence,wehave studied a smart contract on a test network for Bank account of a customer

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

Assignment No. 4

Title: Write a program in solidity to create Student data. Use the following constructs:

- Structures
- Arrays
- Fallback

Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas value.

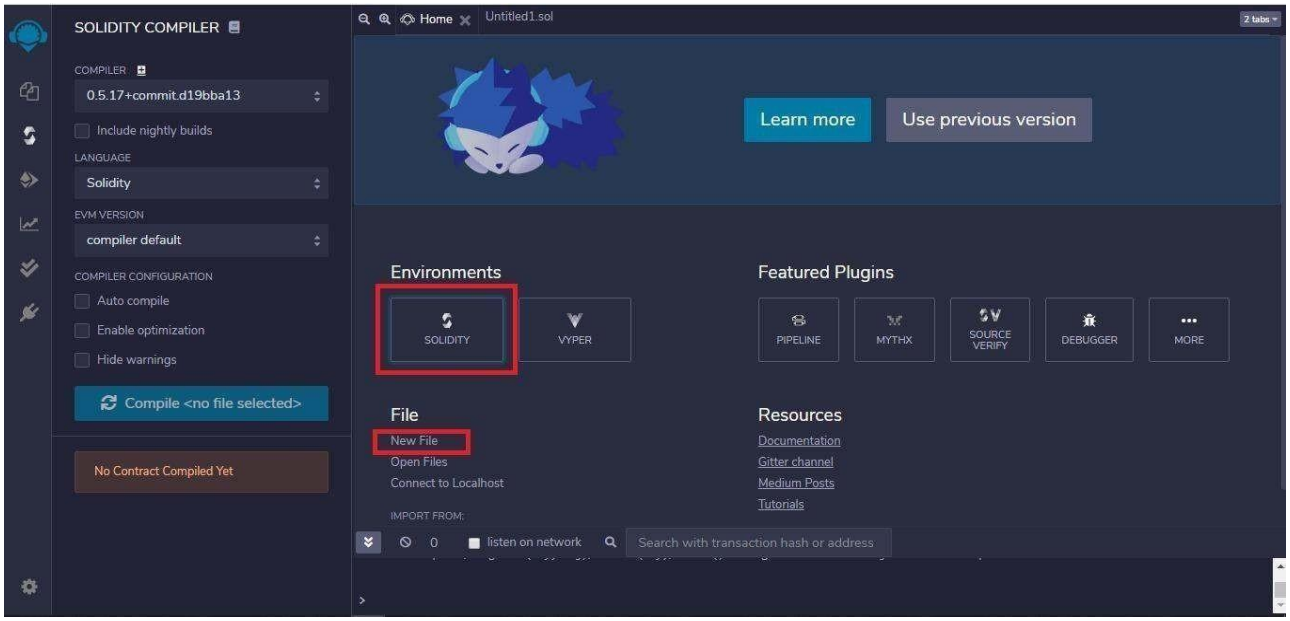
Objective: Understand and explore the working of Blockchain technology and its applications.

Course Outcome:

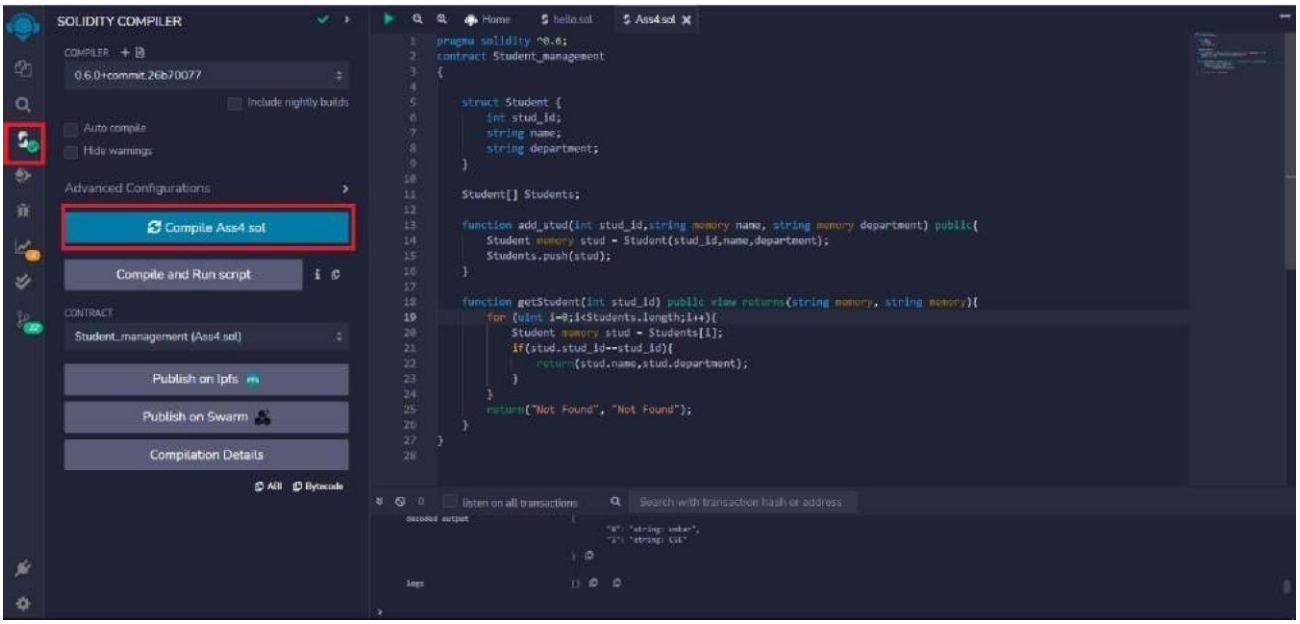
CO6:Interpret the basic concepts in Blockchain technology and its application.

Description:

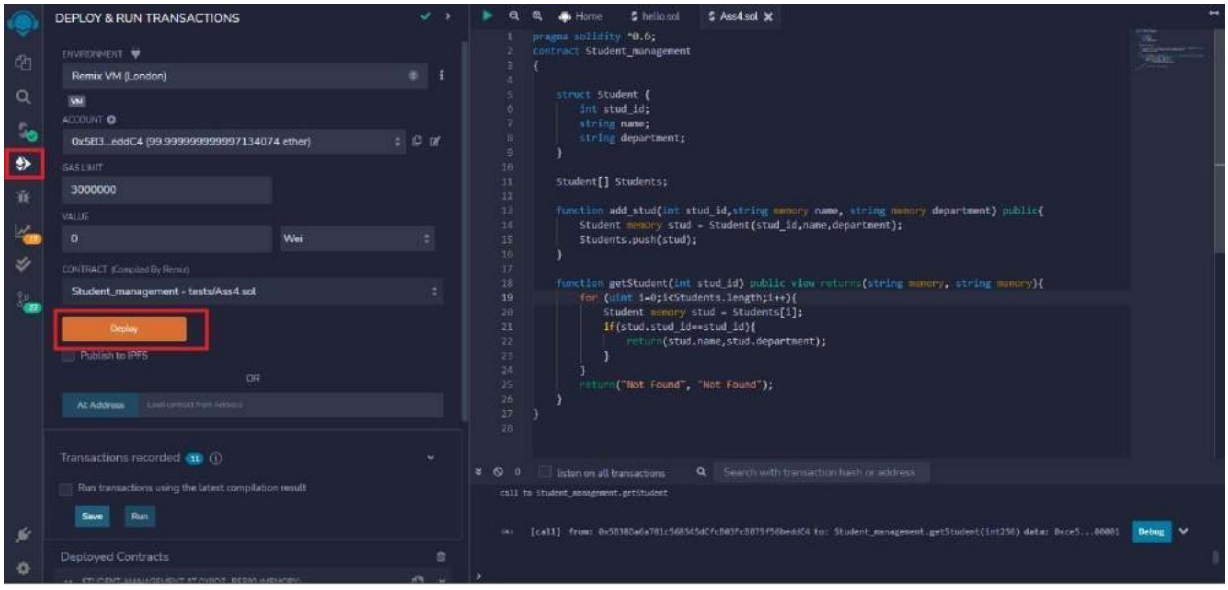
Step 1: Open Remix IDE on any of your browsers, select on the *New File* and click on *Solidity* to choose the environment.



Step 2: Write the Student Management code in the code section, and click the *Compile button* under the Compiler windo w to compile the contract.



Step 3: To execute the code, click on the *Deploy button* under Deploy and Run Transactions window. After deploying the code click on the drop-down on the console.



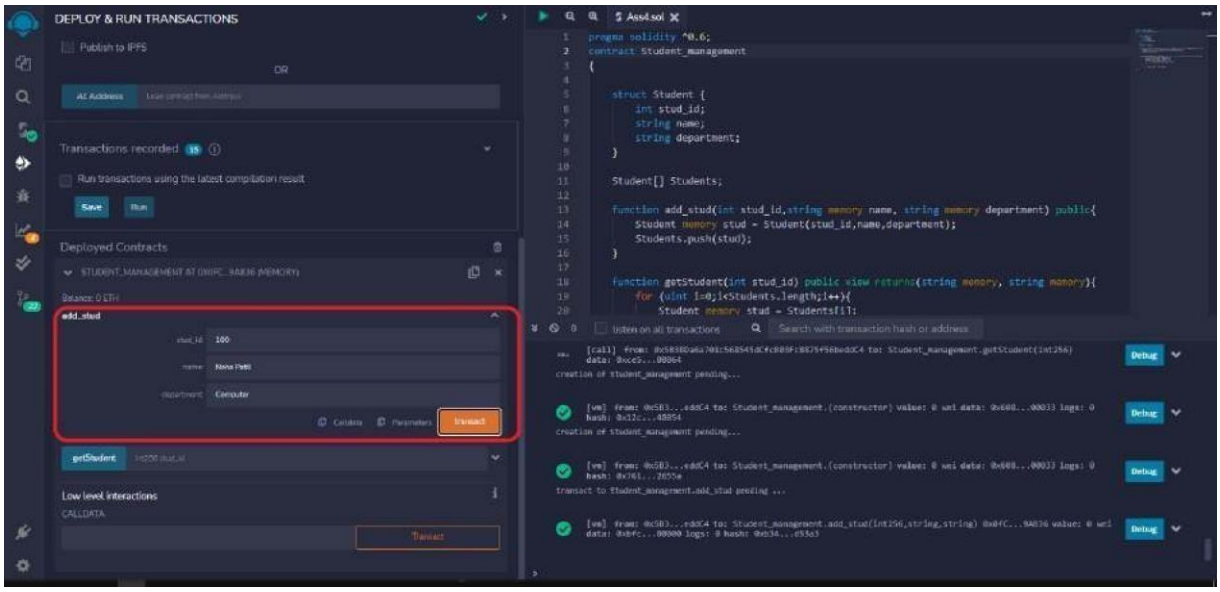
Code

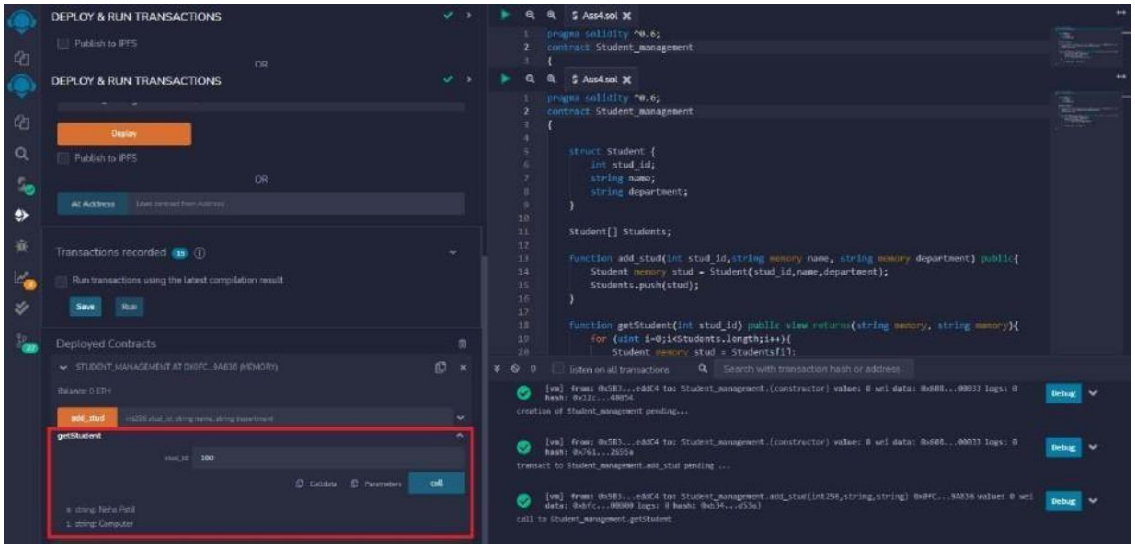
```
pragma solidity ^0.6;
contract Student_management
{

struct Student {
intstud_id;
        string  name;
        string department;
    }

    Student[] Students;

    function add_stud(intstud_id,string memory name, string memory department) public{
        Student memory stud = Student(stud_id,name,department);
Students.push(stud);
    }

    function getStudent(intstud_id) public view returns(string memory, string memory){
        for (uinti=0;i<Students.length;i++){
            Student memory stud = Students[i];
            if(stud.stud_id==stud_id){
                return(stud.name,stud.department);
            }
        }
        return("Not Found", "Not Found");
    }
}
```

Sample Output

After deploying the contact successful you can observe two button add_stud and getStudents.Give the input stud_id, name dept and click on getStudents button, enter the stud_id which you have given as an Input and get the information of Students name and department

Refer the following output





Conclusion: Hence,we have studieda program in solidity to create Student data.

Write-up	Correctness of Program	Documentation o Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

Assignment No : 5

Title of the Assignment: Write a survey report on types of Blockchains and its real time use cases.

Objective of the Assignment: Students should be able to learn new technology such as metamask.Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

Contents for Theory:

There are 4 types of blockchain:

Public

Blockchain.

Private

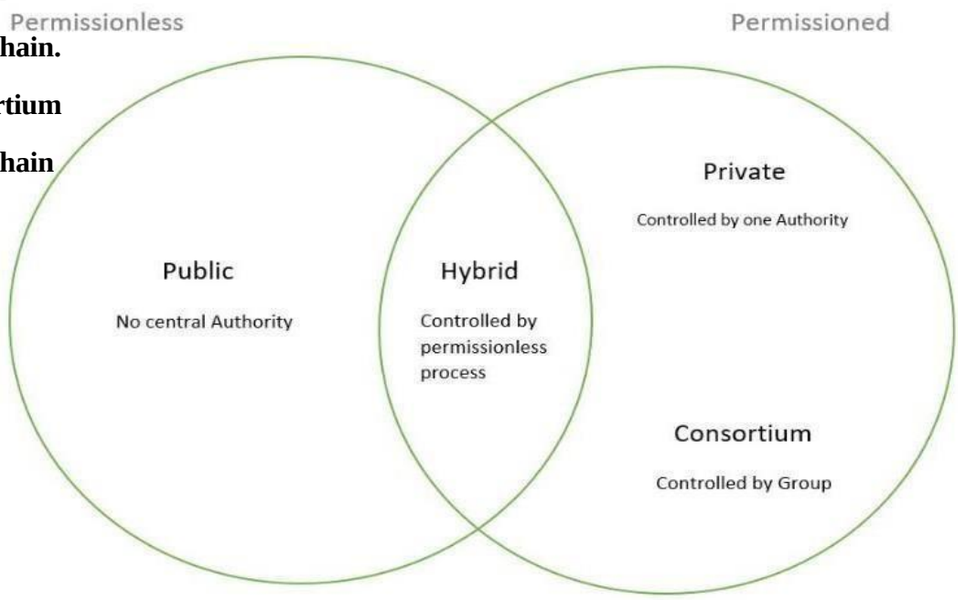
Blockchain.

Hybrid

Blockchain.

Consortium

Blockchain



1. Public Blockchain

These blockchains are completely open to following the idea of decentralization. They don‘t have any restrictions, anyone having a computer and internet can participate in the network.

As the name is public this blockchain is open to the public, which means it is not owned by anyone. Anyone having internet and a computer with good hardware can participate in this public blockchain. All the computer in the network hold the copy of other nodes or block present in the network
In this public blockchain, we can also perform verification of transactions or records
Advantages:

Trustable: There are algorithms to detect no fraud. Participants need not worry about the other nodes in the network
Secure: This blockchain is large in size as it is open to the public. In a large size, there is greater distribution of records
Anonymous Nature: It is a secure platform to make your transaction properly at the same time, you are not required to reveal your name and identity in order to participate.
Decentralized: There is no single platform that maintains the network, instead every user has a copy of the ledger.
Disadvantages:

Processing: The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.
Energy Consumption: Proof of work is high energy-consuming. It requires good computer hardware to participate in the network
Acceptance: No central authority is there so governments are facing the issue to implement the technology faster.
Use Cases: Public Blockchain is secured with proof of work or proof of stake they can be used to displace traditional financial systems. The more advanced side of this blockchain is the smart contract that enabled this blockchain to support decentralization. Examples of public blockchain are Bitcoin, Ethereum.

2. Private Blockchain
These blockchains are not as decentralized as the public blockchain only selected nodes can participate in the process, making it more secure than the others.

These are not as open as a public
blockchain. They are open to some
authorized users only.
These blockchains are operated in a closed network.
In this few people are allowed to participate in a network within a
company/organization. Advantages:

Speed: The rate of the transaction is high, due to its small size. Verification of each node is less time-consuming.
Scalability: We can modify the scalability. The size of the network can be decided manually. Privacy: It has increased the level of privacy for confidentiality reasons as the businesses required.
Balanced: It is more balanced as only some user has the access to the transaction which improves the performance of the network.
Disadvantages:

Security- The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.
Centralized- Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.
Count- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered. Use Cases: With proper security and maintenance, this blockchain is a great asset to secure information without exposing it to the public eye. Therefore companies use them for internal auditing, voting, and asset management. An example of private blockchains is Hyperledger, Corda.

3. Hybrid Blockchain
It is the mixed content of the private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.

It is a combination of both public and private blockchain. Permission-based and permissionless systems are used. User access information via smart contracts
Even a primary entity owns a hybrid blockchain it cannot alter the transaction Advantages:

Ecosystem: Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don‘t have access to the network
Cost: Transactions are cheap as only a few nodes verify the transaction. All the nodes don‘t carry the verification hence less computational cost.
Architecture: It is highly customizable and still maintains integrity, security, and transparency.

Operations: It can choose the participants in the blockchain and decide which transaction can be made public.
Disadvantages:

Efficiency: Not everyone is in the position to implement a hybrid Blockchain. The organization also faces
some difficulty in terms of efficiency in maintenance.
Transparency: There is a possibility that someone can hide information from the user. If someone wants to get access through a hybrid blockchain it depends on the organization whether they will give or not.
Ecosystem: Due to its closed ecosystem this blockchain lacks the incentives for network participation.
Use Case: It provides a greater solution to the health care industry, government, real estate, and financial companies. It provides a remedy where data is to be accessed publicly but needs to be shielded privately. Examples of Hybrid Blockchain are Ripple network and XRP token.

4. Consortium Blockchain
It is a creative approach that solves the needs of the organization. This blockchain validates the transaction and also initiates or receives transactions.

Also known as Federated Blockchain.
This is an innovative method to solve the organization's
needs. Some part is public and some part is private.
In this type, more than one organization manages the
blockchain. Advantages:

Speed: A limited number of users make verification fast. The high speed makes this more usable for organizations.
Authority: Multiple organizations can take part and make it decentralized at every level.
Decentralized authority, makes it more secure.
Privacy: The information of the checked blocks is unknown to the public view. but any member belonging to the blockchain can access it.
Flexible: There is much divergence in the flexibility of the blockchain. Since it is not a very large decision can be taken faster.
Disadvantages:

Approval: All the members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.
Transparency: It can be hacked if the organization becomes corrupt. Organizations may hide information from the users.
Vulnerability: If few nodes are getting compromised there is a greater chance of vulnerability in this blockchain
Use Cases: It has high potential in businesses, banks, and other payment processors. Food tracking of the organizations frequently collaborates with their sectors making it a federated solution ideal for their use.
Examples of consortium Blockchain are Tendermint and Multichain.

Conclusion-In this way we have explored types of blockchain and its applications in real time

Write-up	Correctness of Program	Documentation o Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

Assignment No : 6

Title of the Assignment: Write a program to create a Business Network using Hyperledger.

Objective of the Assignment: Students should be able to learn hyperledger .Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

Contents for Theory:

Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier. The primary goal is to accelerate time to value, and make it easier to integrate your blockchain applications with the existing business systems.

- You can use Composer to rapidly develop use cases and deploy a blockchain solution in days.
- Composer allows you to model your business network and integrate existing systems and data with your blockchain applications.
- Hyperledger Composer supports the existing Hyperledger Fabric blockchain infrastructure and runtime.
- Hyperleder Composer generate business network archive (bna) file which you can deploy on existing Hyperledger Fabric network

You can use Hyperledger Composer to model business network, containing your existing assets and the transactions related to them

Key Concepts of Hyperledger Composer

1. Blockchain State Storage: It stores all transaction that happens in your hyperledger composer application. It stores transaction in Hyperledger fabric network.

2. Connection Profiles: Connection Profiles to configuration JSON file which help composer to connect to Hyperledger Fabric. You can find Connection Profile JSON file in user‘s home directory.

3. Assets: Assets are tangible or intangible goods, services, or property, and are stored in registries. Assets can represent almost anything in a business network, for example, a house for sale, the sale listing, the land registry certificate for that house. Assets must have a unique identifier, but other than that, they can contain whatever properties you define.

4. Participants: Participants are members of a business network. They may own assets and submit transactions. Participant must have an identifier and can have any other properties.

5. Identities and ID cards: Participants can be associated with an identity. ID cards are a combination of an identity, a connection profile, and metadata. ID cards simplify the process of connecting to a business network.

6. Transactions: Transactions are the mechanism by which participants interact with assets. Transaction processing logic you can define in JavaScript and you can also emit event for transaction.

7. Queries: Queries are used to return data about the blockchain world-state. Queries are defined within a business network, and can include variable parameters for simple customisation. By using queries, data can be easily extracted from your blockchain network. Queries are sent by using the Hyperledger Composer API.

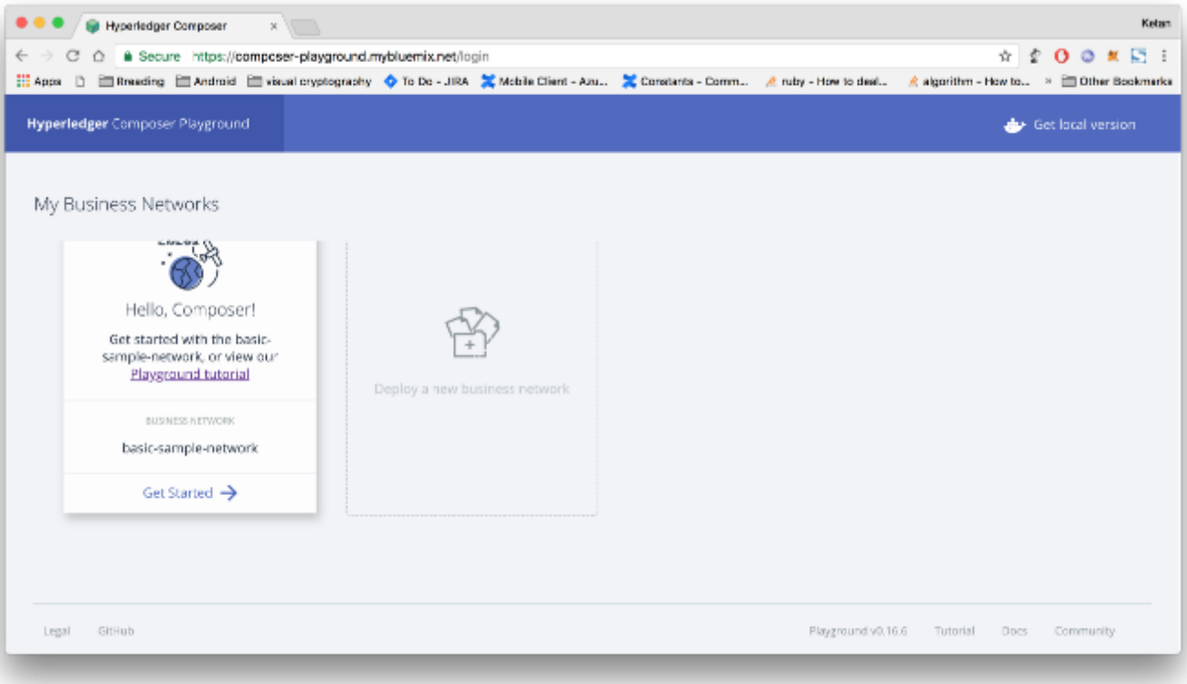
8. Events: Events are defined in the model file. Once events have been defined, they can be emitted by transaction processor functions to indicate to external systems that something of importance has happened to the ledger.

9. Access Control: Hyperledger is enterprise blockchain and access control is core feature of any business blockchain. Using Access Control rules you can define who can do what in Business networks. The access control language is rich enough to capture sophisticated conditions.

10. Historian registry: The historian is a specialised registry which records successful transactions, including the participants and identities that submitted them. The historian stores transactions as HistorianRecord assets, which are defined in the Hyperledger Composer system namespace.

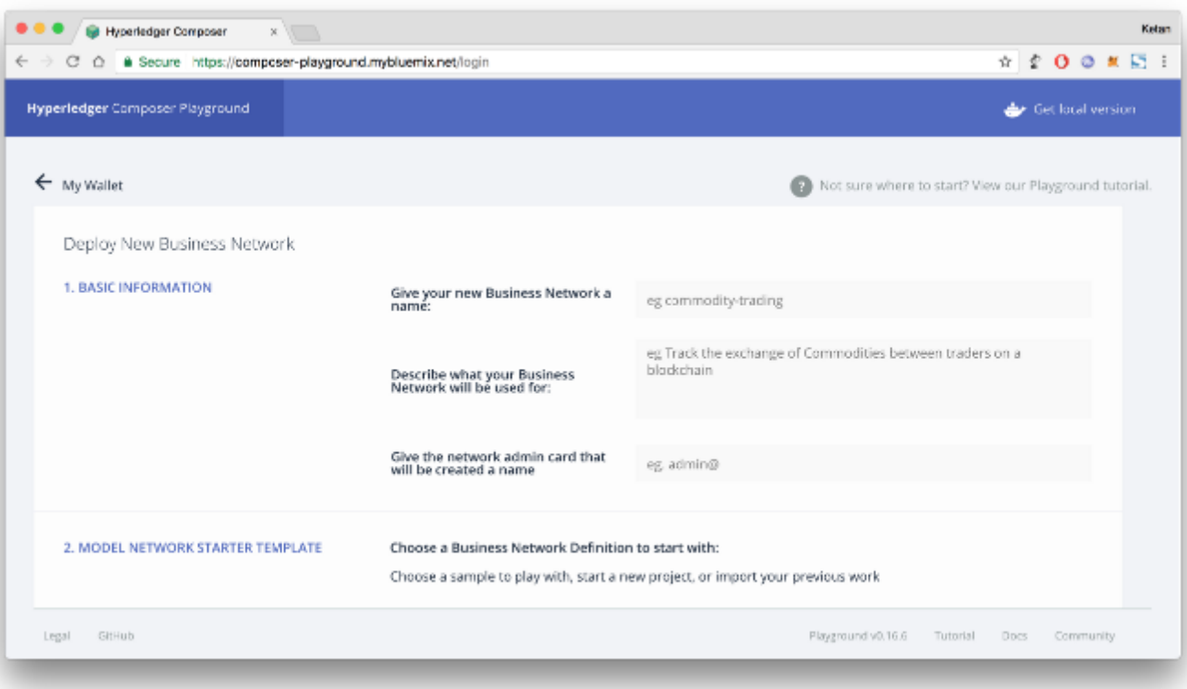
Let’s create first Hyperledger Composer Application

Step 1: Start Hyperledger Composer Online version of Local. Click on Deploy a new business network

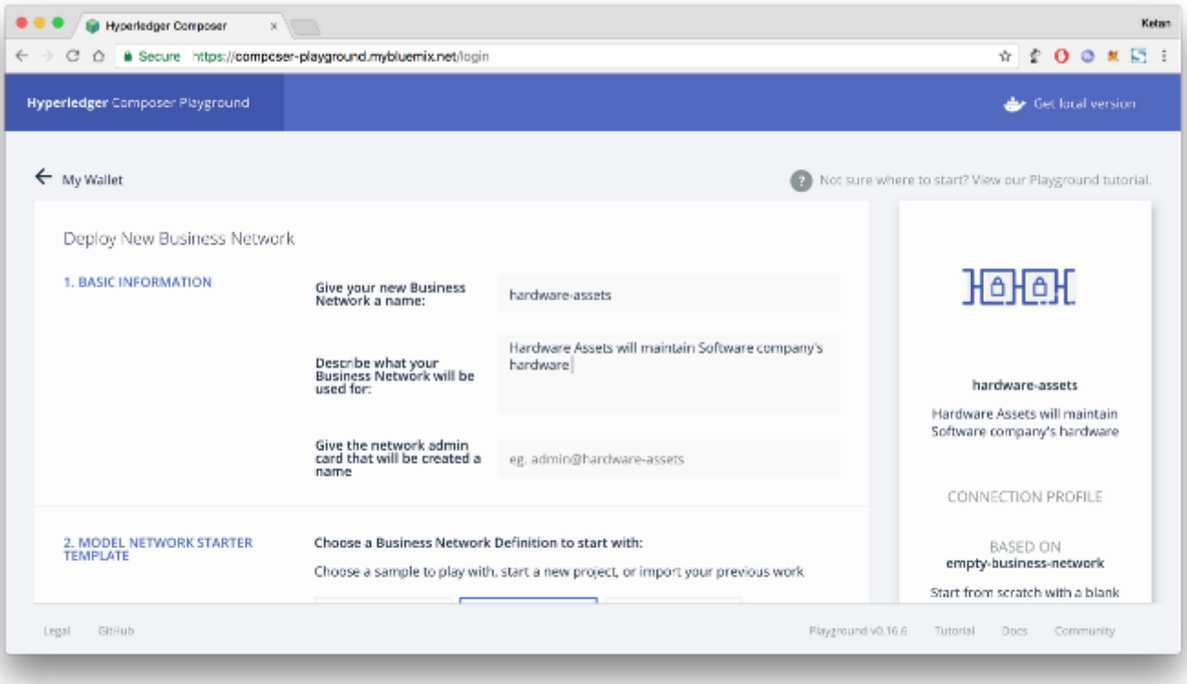


Hyperledger Composer Playground Online version

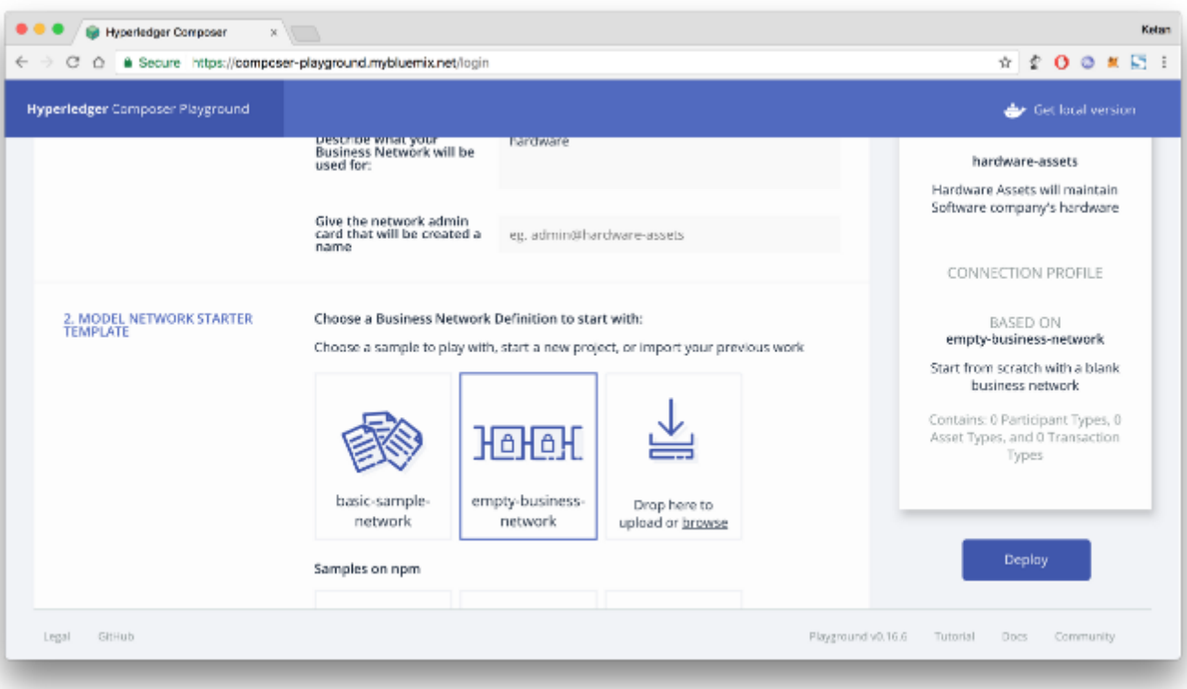
Step 2: Select empty business network



Step 3: Fill basic information, select empty business network and click ¯deploy‖ button from right pannel

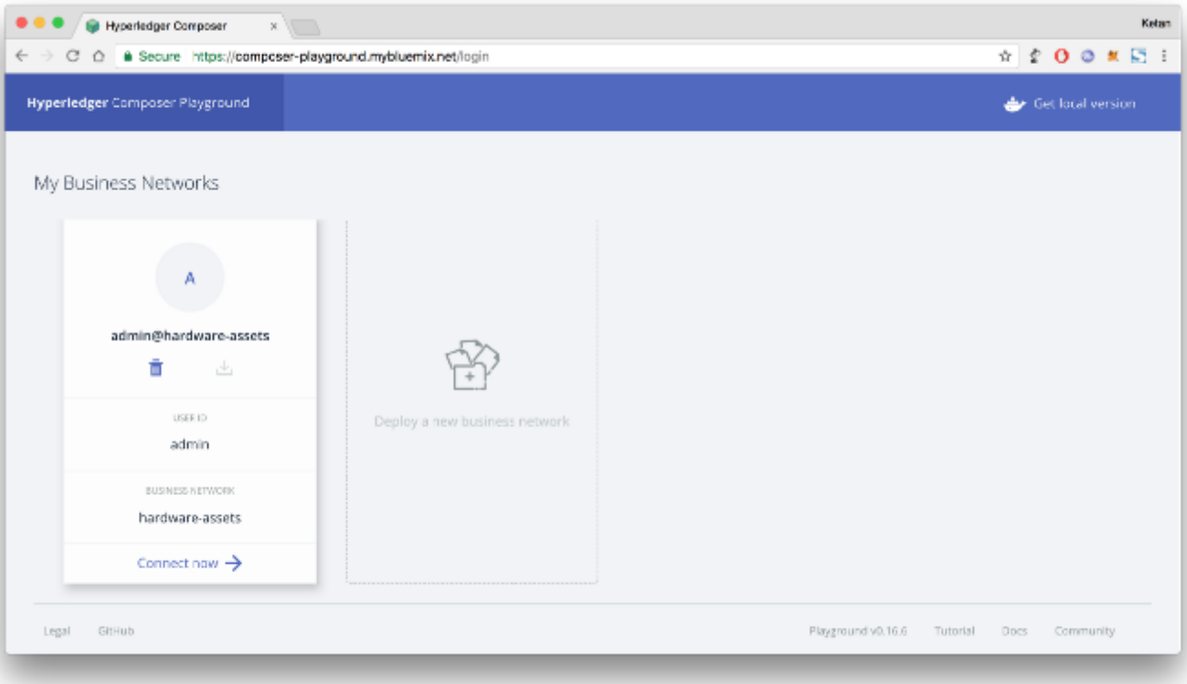


Fill basic information

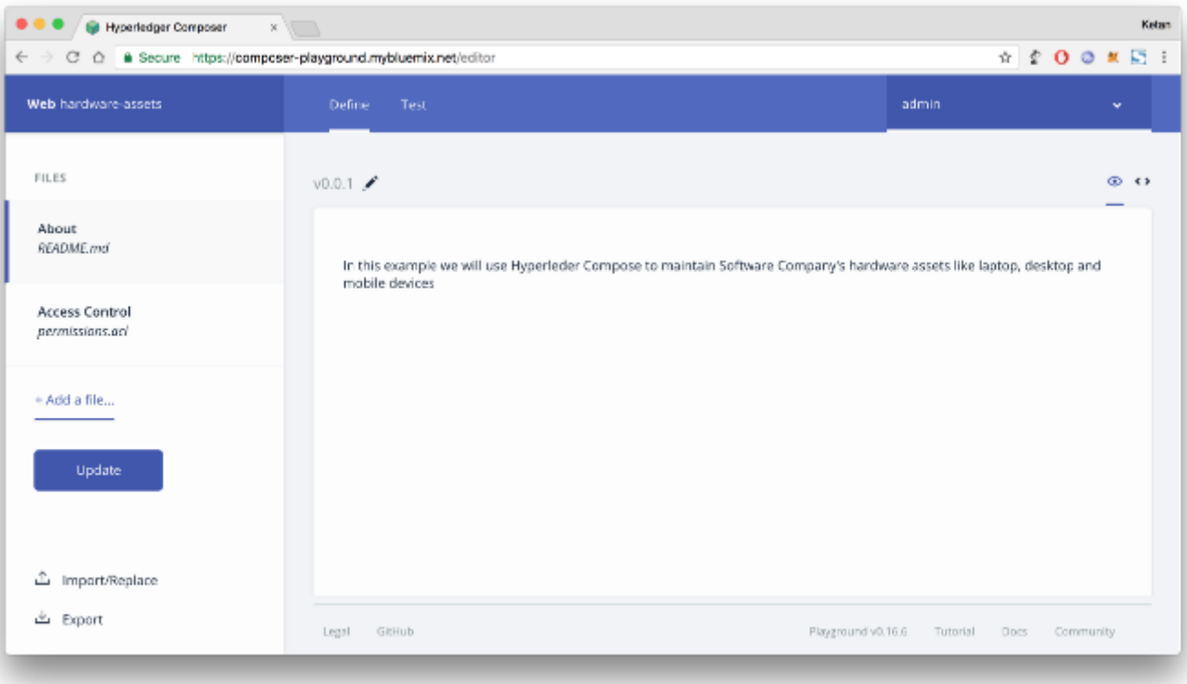


select empty business network

Step 4: Connect to ¯hardware-assets‖ business network that we have just deploied

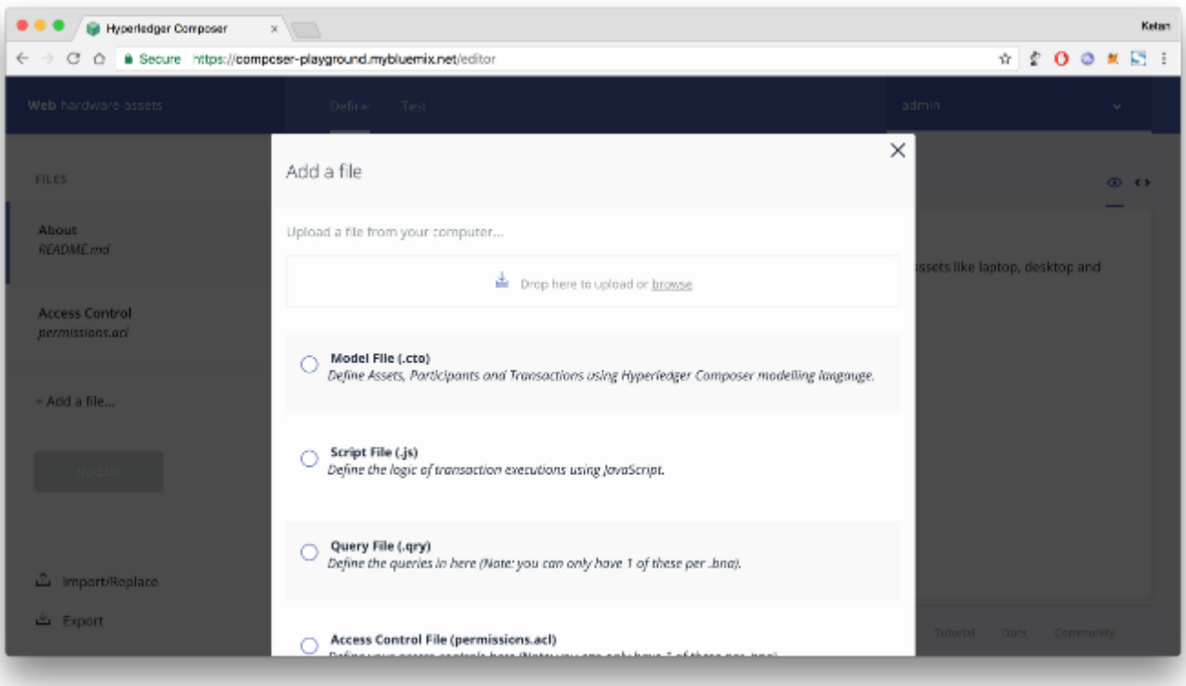


click on ¯connect now‖ button



Inside hardware-assets business network

Step 5: Click on ¯+Add a file...‖ from left panel and select ¯model file (.cto)‖



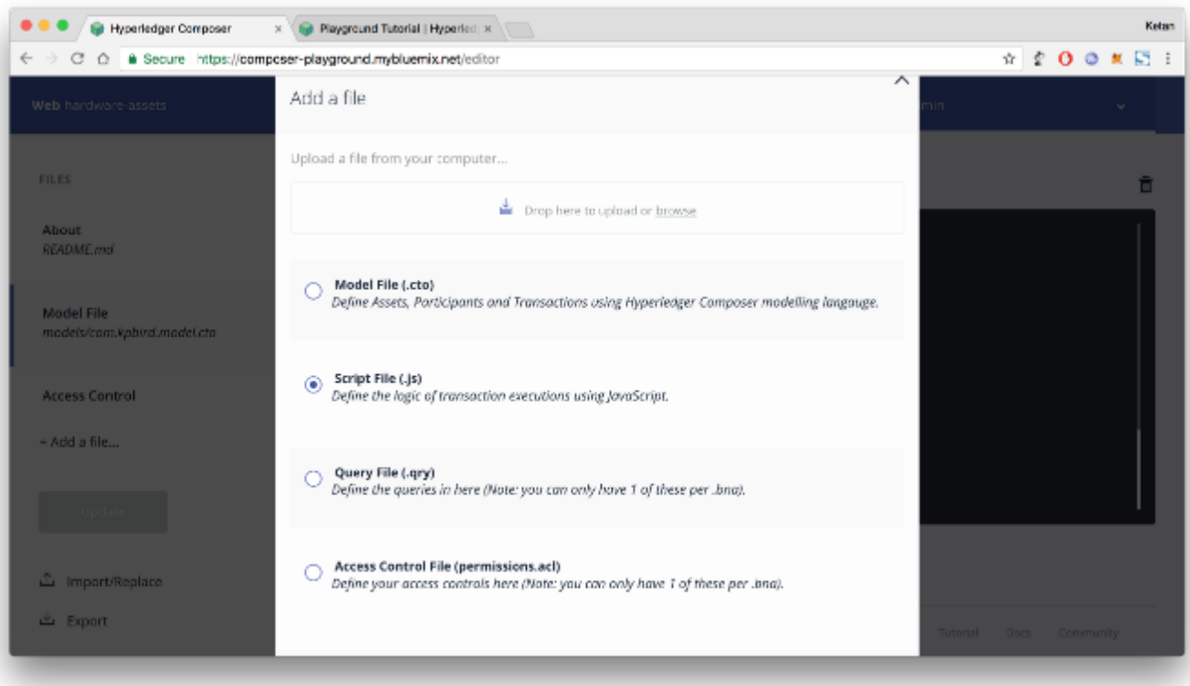
Write following code in model file. Model file contain asset in our case it‘s hardware, participant in our case participants are employee of organisation and transaction as Allocate hardware to employee. Each model has extra properties. Make sure your have proper and unique namespace. In this example I am using ¯com.kpbird‖ as namespace. You can access all models using this namespace i.e. com.kpbird.Hardware, com.kpbird.Employee

```
/**
 * Hardware model
 */namespace com.kpbirdasset Hardware identified by hardwareId {
 o String hardwareId
 o String name
 o String type
 o String description
 o Double quantity
 → Employee owner
}
participant Employee identified by employeeId {
 o String employeeId
 o String firstName
 o String lastName
}
transaction Allocate {
 → Hardware hardware
 → Employee newOwner
}
```

Hyperledger modeling language

reference: https://hyperledger.github.io/composer/reference/cto_language.html

Step 6: Click on ¯+Add a file...‖ from left panel and select ¯script file (*.js)‖



Write following code in Script File. In Script we can define transaction processing logic. In our case we want to allocate hardware to the employee so, we will update owner of hardware. Make sure about annotation above functions @params and @transaction

```
/**
 * Track the trade of a commodity from one trader to another
 * @param {com.kpbird.Allocate} trade – the trade to be processed
 * @transaction
 */
function allocateHardware(allocate) {
 allocate.hardware.owner = allocate.newOwner;
 return getAssetRegistry(_com.kpbird.Hardware‘)
 .then(function (assetRegistry) {
 return assetRegistry.update(allocate.hardware);
 });
}
```

Hyperledger Composer Script file reference: https://hyperledger.github.io/composer/reference/js_scripts.html

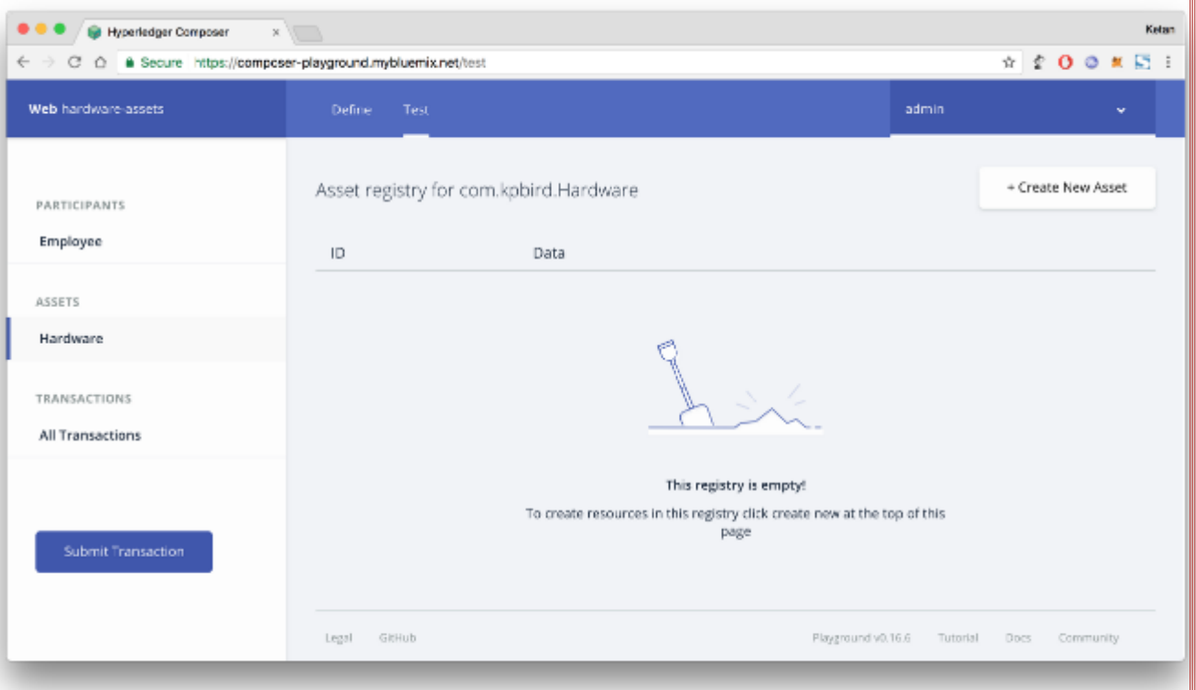
Step 7: permissions.acl file sample is already available, Add following code in permissions.acl file.

```
/**
 * New access control file
 */
 rule AllAccess {
 description: ¯AllAccess – grant everything to everybody.‖
 participant: ¯ANY‖
 operation: ALL
 resource: ¯com.kpbird.**‖
 action: ALLOW
 }rule SystemACL{
 description: ¯System ACL to permit all access‖
 participant: ¯org.hyperledger.composer.system.Participant‖
 operation: ALL
 resource: ¯org.hyperledger.composer.system.**‖
 action: ALLOW
}
```

Hyperledger Composer Access Control Language

reference: https://hyperledger.github.io/composer/reference/acl_language.html

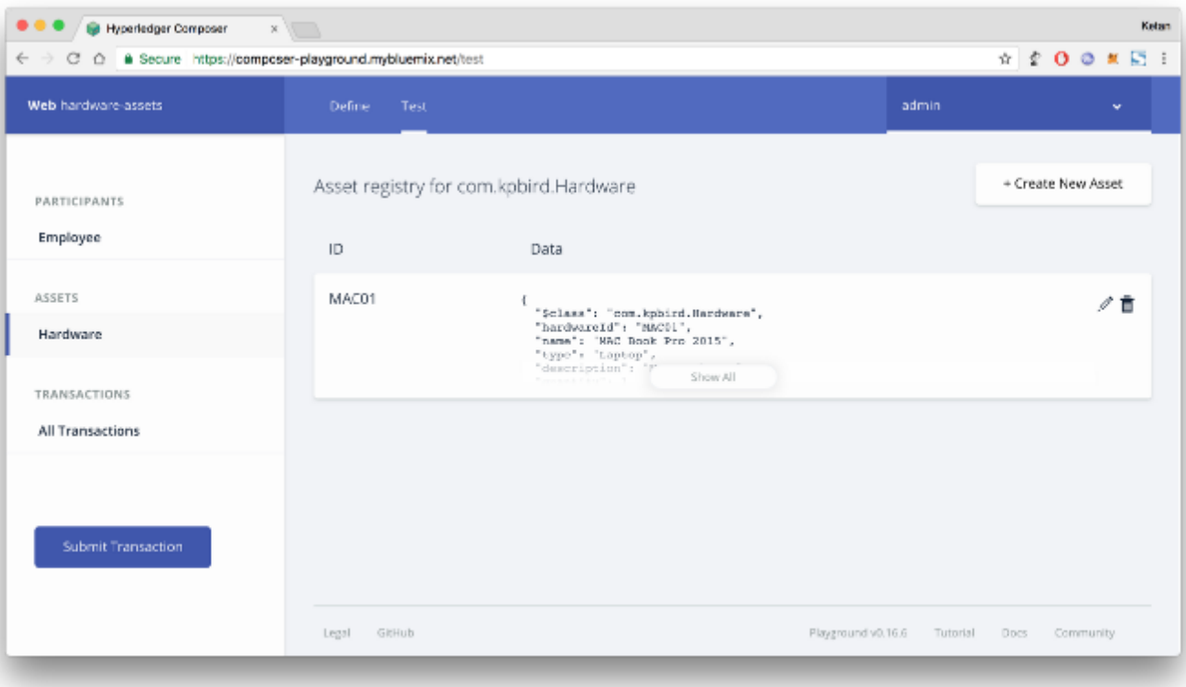
Step 8: Now, It's time to test our hardware assets business network. Hyperledger composer gives ¯Test‖ facility from composer panel it self. Click on ¯Test‖ tab from top panel



Test feature of Hyperledger Composer

Step 9: Create Assets. Click on ¯Hardware‖ from left panel and click ¯+ Create New Assets‖ from right top corner and add following code. We will create Employee#01 in next step. Click on ¯Create New‖ button

```
{
  ¯$class‖: ¯com.kpbird.Hardware‖,
  ¯hardwareId‖: ¯MAC01‖,
  ¯name‖: ¯MAC Book Pro 2015‖,
  ¯type‖: ¯Laptop‖,
  ¯description‖: ¯Mac Book Pro‖,
  ¯quantity‖: 1,
  ¯owner‖: ¯resource:com.kpbird.Employee#01‖
}
```



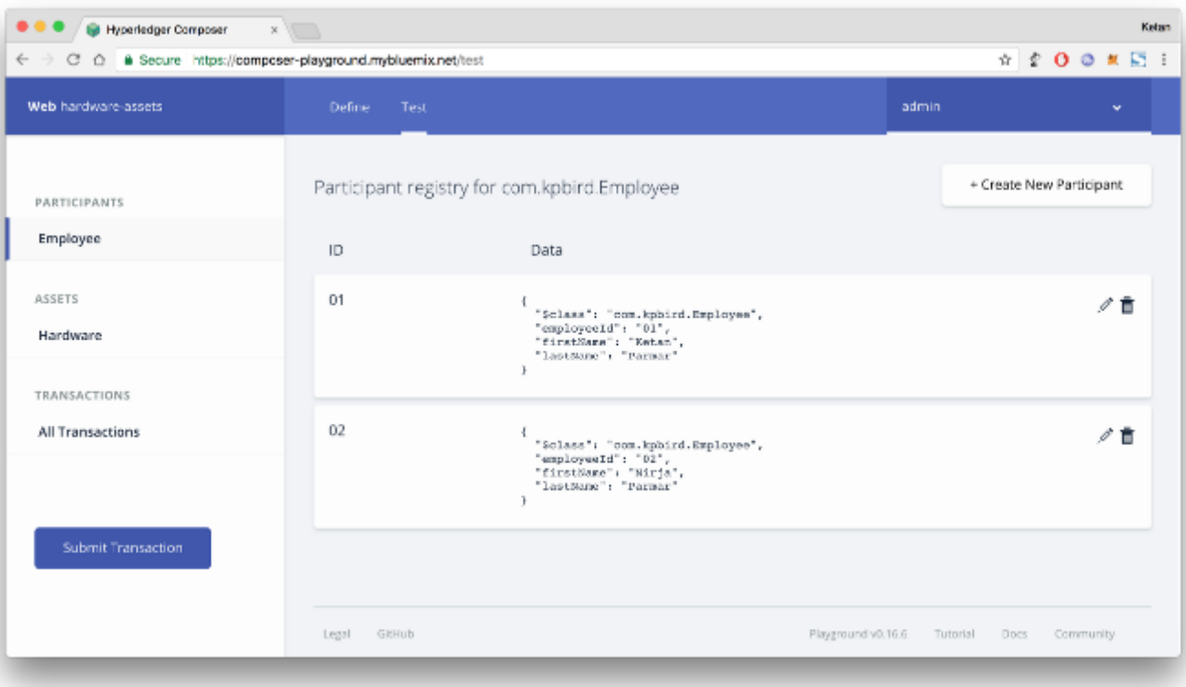
After adding Hardware assets

Steps 10: Let's create participants. Click "Employee" and click "+ Create New Participants" and add following code. We will add two employees

```
{
  "$class": "com.kpbird.Employee",
  "employeeId": "01",
  "firstName": "Ketan",
  "lastName": "Parmar"
}
```

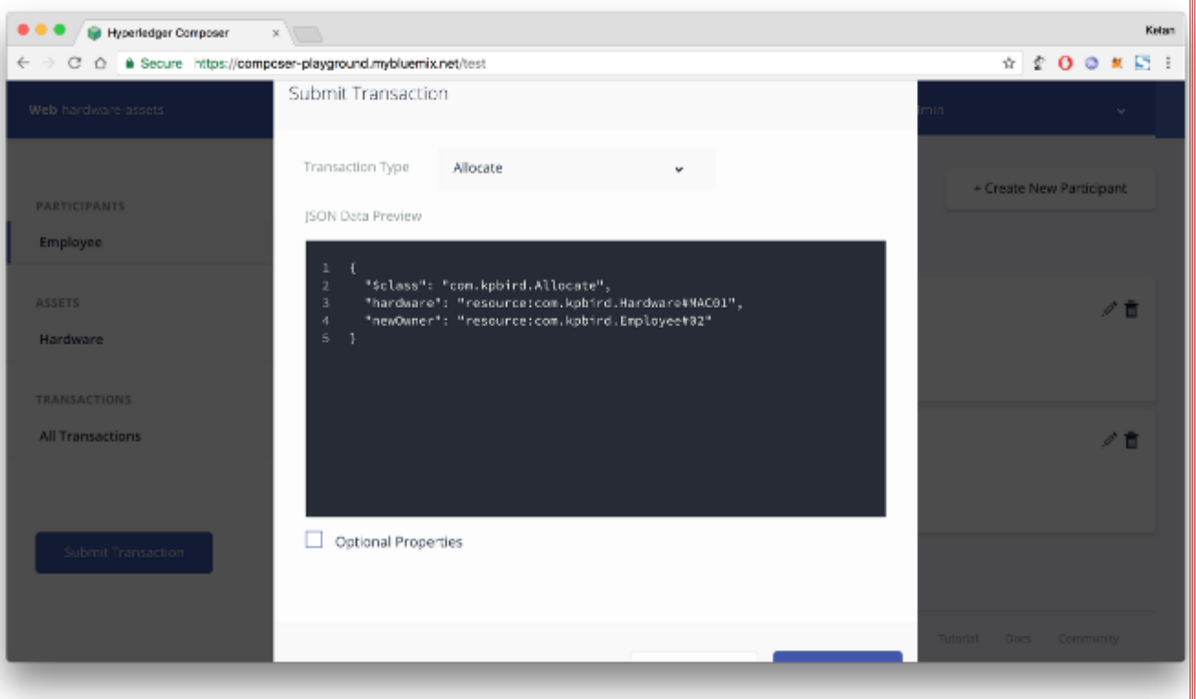
Click on "Create New" on dialog

```
{
  "$class": "com.kpbird.Employee",
  "employeeId": "02",
  "firstName": "Nirja",
  "lastName": "Parmar"
}
```



We have two employees

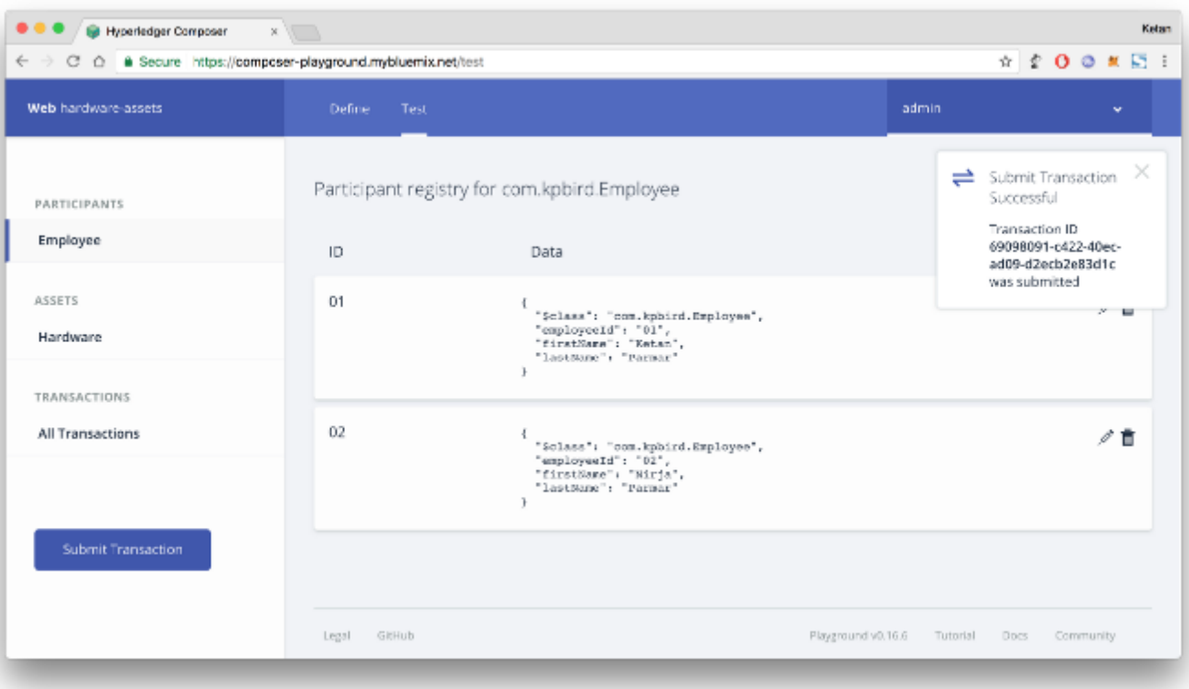
Step 11: It‘s time to do transaction, We will allocate Macbook Pro from Ketan (Employee#01) to Nirja (Employee#02). Click on ¯Submit Transaction‖ button from left panel. In Transaction dialog, We can see all transaction functions on top ¯Transaction Type‖ dropdown.



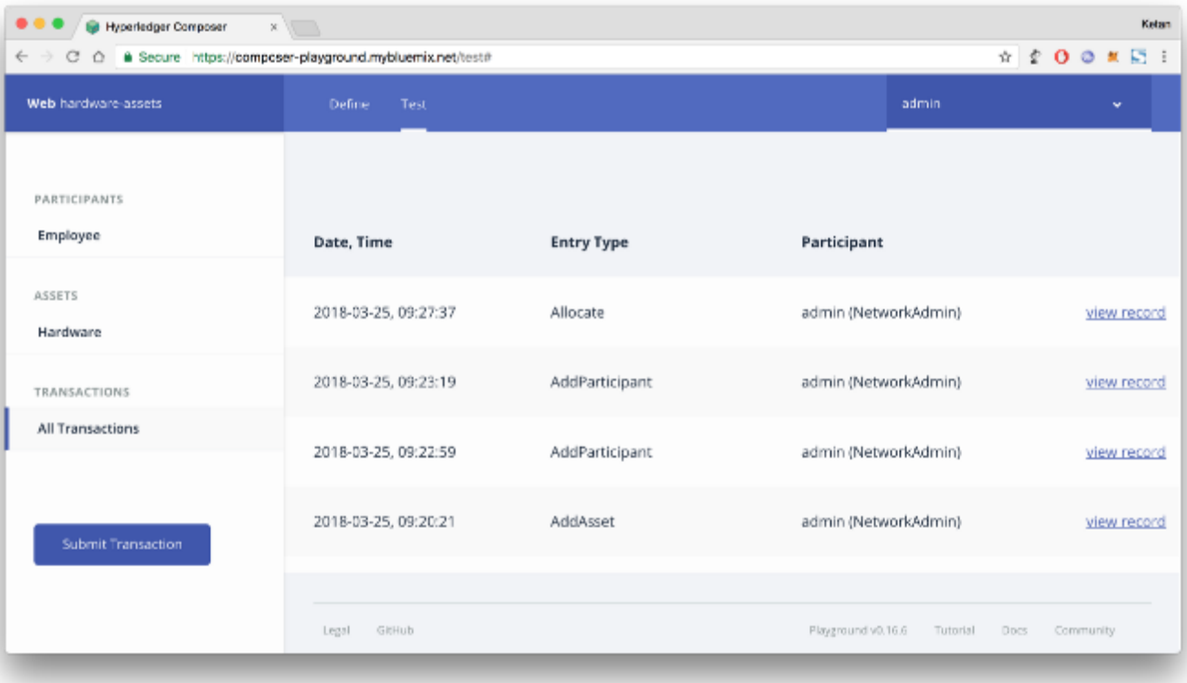
Submit Transaction Dialog

```
{
 ¯$class‖: ¯com.kpbird.Allocate‖,
 ¯hardware‖: ¯resource:com.kpbird.Hardware#MAC01‖,
 ¯newOwner‖: ¯resource:com.kpbird.Employee#02‖
}
```

Now, We are allocating Mac01 to Employee 02. Click Submit button after update above JSON in Transaction Dialog. As soon as you hit submit button. Transaction processed and Transaction Id will generate.

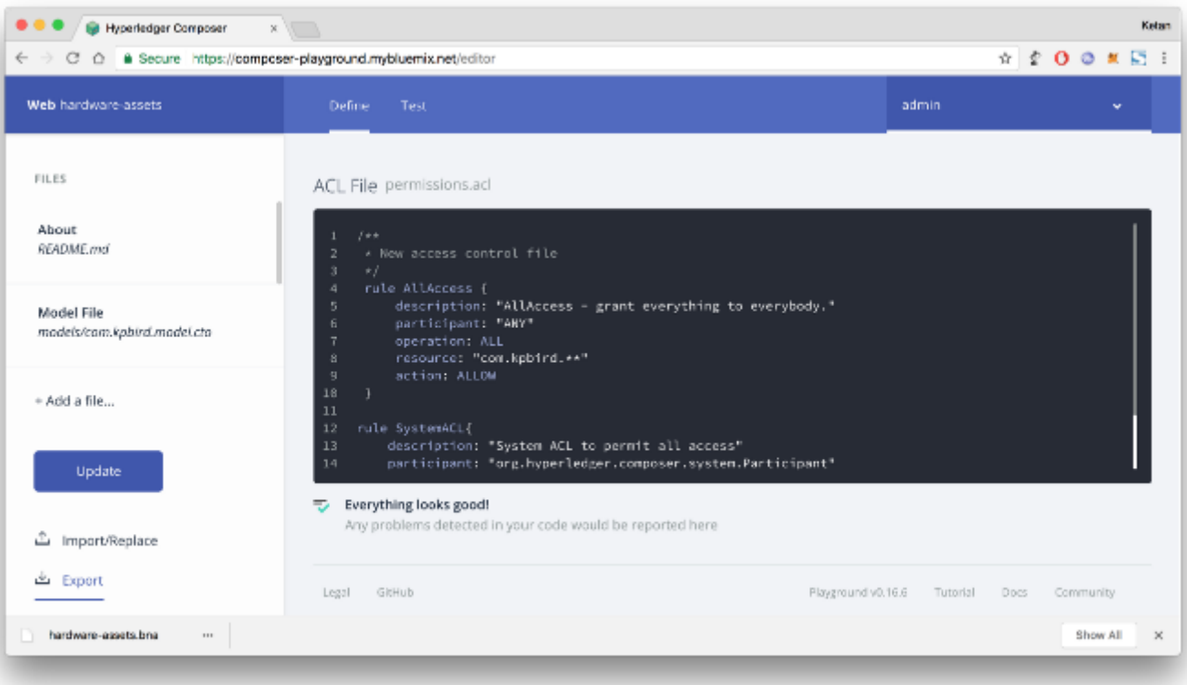


Step 12: Click on ―All Transactions‖ from left panel to verify all transactions. In following screenshots you can see add assets, ass participants and allocation all operation are consider as transactions. ―view records‖ will give us more information about transaction.



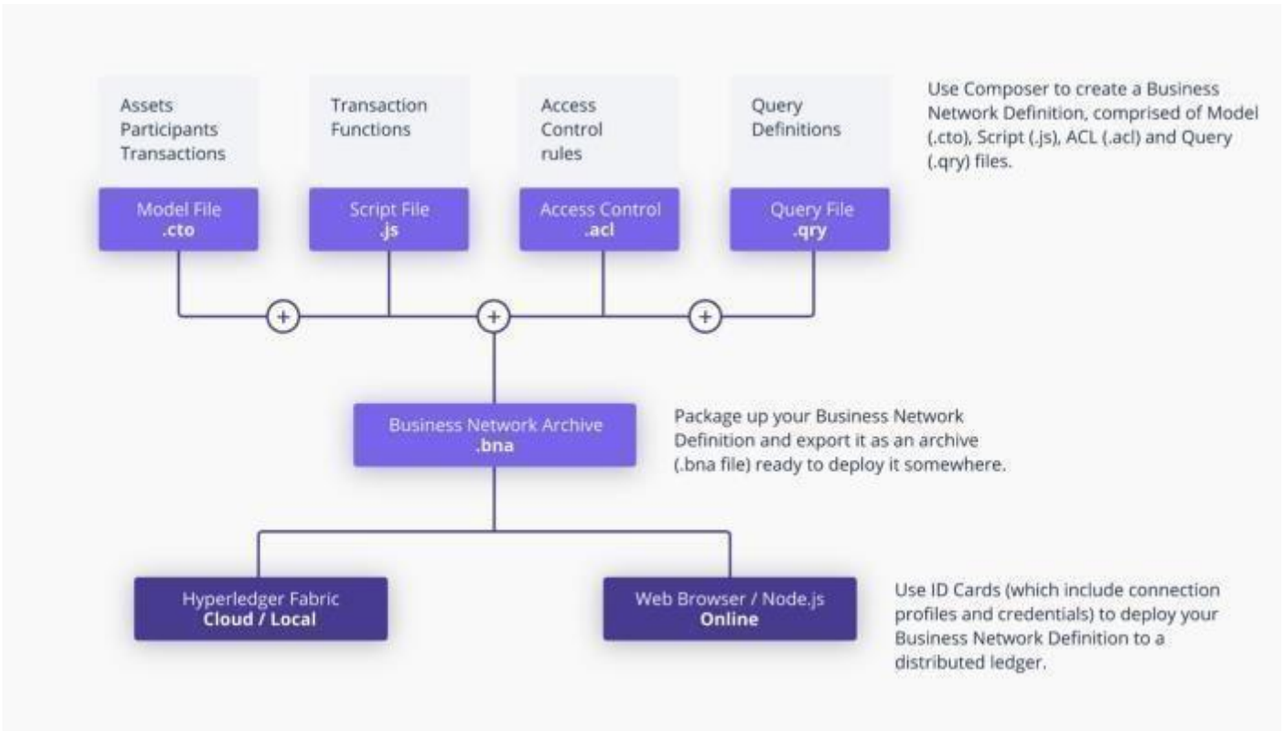
All Transactions

Step 13: Now, It‘s time to deploy ―hardware-assets‖ business network to Hyperledger Fabric. Click on ―Define‖ tab from top panel and click ―Export‖ button from left panel. Export will create hardware-assets.bna file.



Download hardware-assets.bna file

.bna is Business Network Archive file which contains model, script, network access and query file



source: https://hyperledger.github.io/composer/introduction/introduction

Step 14: Start Docker and run following commands from ~/fabric-tools directory

Install business network to Hyperledger Fabric, If business network is already installed you can use ¯update‖ instead of ¯install‖

```
$composer runtime install -c PeerAdmin@hlfv1 -n hardware-assets
```

```
Ketan-Parmar:fabric-tools ketan$ composer runtime install -c PeerAdmin@hlfv1 -n hardware-assets
✔ Installing runtime for business network hardware-assets. This may take a minute...

Command succeeded
```

Following command will deploy and start hardware-assets.bna file. Change hardware-assets.bna file before you execute following command. networkadmin.card file will generate in ~/fabric-tools directory from previous command.

```
$composer network start – card PeerAdmin@hlfv1 – networkAdmin admin – networkAdminEnrollSecret adminpw – archiveFile /Users/ketan/Downloads/hardware-assets.bna – file networkadmin.card
```

```
Ketan-Parmar:fabric-tools ketan$ composer network start --card PeerAdmin@hlfv1 --networkAdmin admin --networkAdminEnrollSecr
et adminpw --archiveFile /Users/ketan/Downloads/hardware-assets.bna --file networkadmin.card
Starting business network from archive: /Users/ketan/Downloads/hardware-assets.bna
Business network definition:
        Identifier: hardware-assets@0.0.1
        Description: Hardware Assets will maintain Software company's hardware

Processing these Network Admins:
        userName: admin

✔ Starting business network definition. This may take a minute...
Successfully created business network card:
        Filename: networkadmin.card

Command succeeded
```

To connect business network you need connection card. so we can import networkadmin.card using following command

```
$composer card import -f networkadmin.card
```

To make sure networkadmin.card successfully install you can list cards using following command

```
$composer card list
```

```
Ketan-Parmar:fabric-tools ketan$ composer card list
The following Business Network Cards are available:

Connection Profile: hlfv1
```

Card Name	UserId	Business Network
admin@hardware-assets	admin	hardware-assets
PeerAdmin@trade-network	PeerAdmin	trade-network
admin@trade-network	admin	trade-network
PeerAdmin@hlfv1	PeerAdmin	

```
Issue composer card list --name <Card Name> to get details a specific card

Command succeeded
```

Following command will make sure that our hardware-assets business network is successfully running in Hyperledger Fabric.

$composer network ping – card admin@hardware-assets

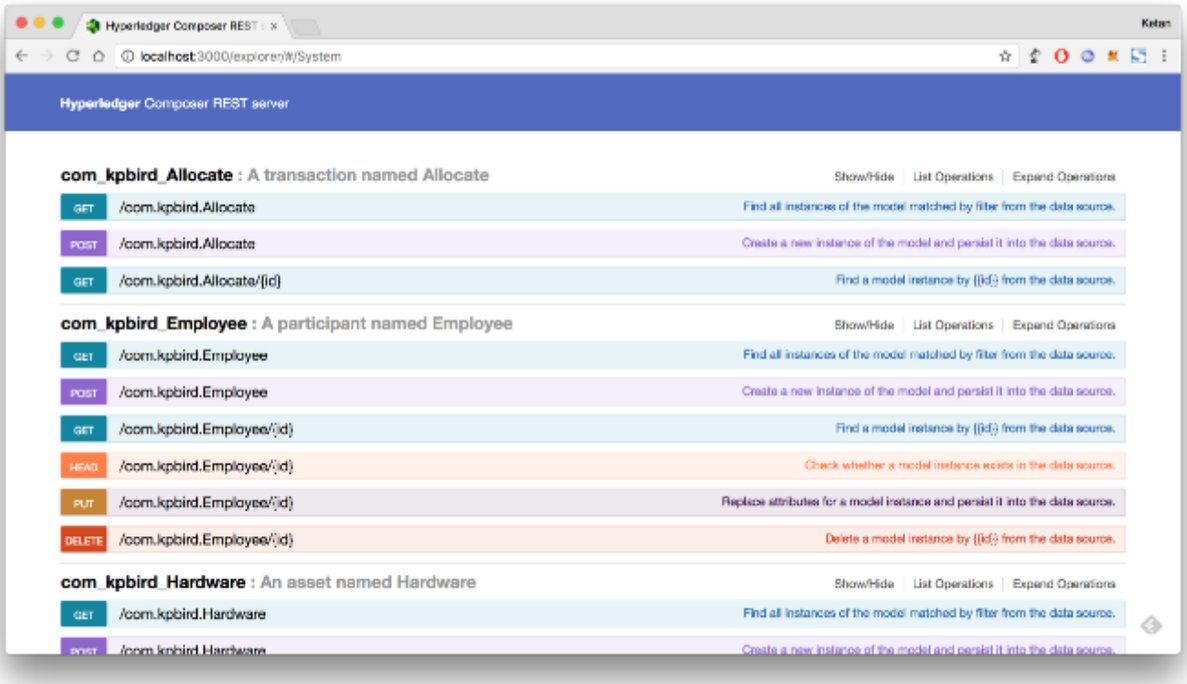
```
Ketan-Parmar:fabric-tools ketan$ composer network ping --card admin@hardware-assets
The connection to the network was successfully tested: hardware-assets
        version: 0.16.0
        participant: org.hyperledger.composer.system.NetworkAdmin#admin

Command succeeded
```

Now It's time to interact with REST API. To develop Web or Mobile Application we require REST API. you can run following command to generate REST API for hardware-assets business network.

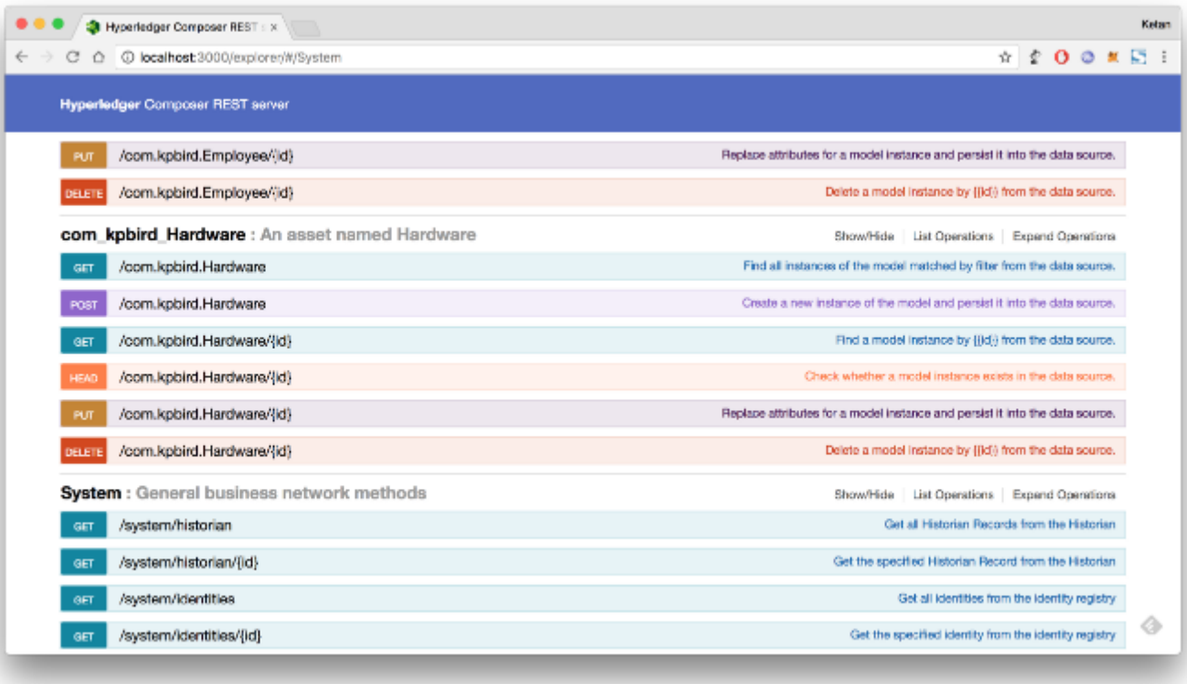
$composer-rest-server

```
Ketan-Parmar:fabric-tools ketan$ composer-rest-server
? Enter the name of the business network card to use: admin@hardware-assets
? Specify if you want namespaces in the generated REST API: always use namespaces
? Specify if you want to enable authentication for the REST API using Passport: No
? Specify if you want to enable event publication over WebSockets: Yes
? Specify if you want to enable TLS security for the REST API: No

To restart the REST server using the same options, issue the following command:
   composer-rest-server -c admin@hardware-assets -n always -w true

Discovering types from business network definition ...
Discovered types from business network definition
Generating schemas for all types in business network definition ...
Generated schemas for all types in business network definition
Adding schemas for all types to Loopback ...
Added schemas for all types to Loopback
Web server listening at: http://localhost:3000
Browse your REST API at http://localhost:3000/explorer
```

rest server will ask few basic information before generate rest api



REST API for our hardware assets



REST API methods for all operations

Conclusion: In this way we have learnt about hyperledger and its use case in business world.

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

MINI PROJECT-3

Problem Statement: Develop a Blockchain based application dApp (de- centralized app) for evoting system.

Theory:

Blockchain is a technology that is rapidly gaining momentum in era of industry 4.0. With high security and transparency provisions, it is being widely used in supply chain management systems, healthcare, payments, business, IoT, voting systems, etc.

Why do we need it?

Current voting systems like ballot box voting or electronic voting suffer from various security threats such as DDoS attacks, polling booth capturing, vote alteration and manipulation, malware attacks, etc, and also require huge amounts of paperwork, human resources, and time. This creates a sense of distrust among existing systems. Some of the disadvantages are:

- Long Queues during elections.
- Security Breaches like data leaks, vote tampering.
- Lot of paperwork involved, hence less eco-friendly and time-consuming.
- Difficult for differently-abled voters to reach polling booth.
- Cost of expenditure on elections is high.

Solution:

Using blockchain, voting process can be made more secure, transparent, immutable, and reliable. How? Let‘s take an example.

Suppose you are an eligible voter who goes to polling booth and cast vote using EVM (Electronic Voting Machine). But since it‘s a circuitry after all and if someone tampers with microchip, you may never know that did your vote reach to person for whom you voted or was diverted into another candidate‘s account?

Since there‘s no tracing back of your vote. But, if you use blockchain- it stores everything as a transaction that will be explained soon below; and hence gives you a receipt of your vote (in a form of a transaction ID) and you can use it to ensure that your vote has been counted securely.

Now suppose a digital voting system (website/app) has been launched to digitize process and all confidential data is stored on a single admin server/machine, if someone tries to hack it or snoop over it, he/she can change candidate's vote count- from 2 to 22! You may never know that hacker installs malware or performs clickjacking attacks to steal or negate your vote or simply attacks central server.

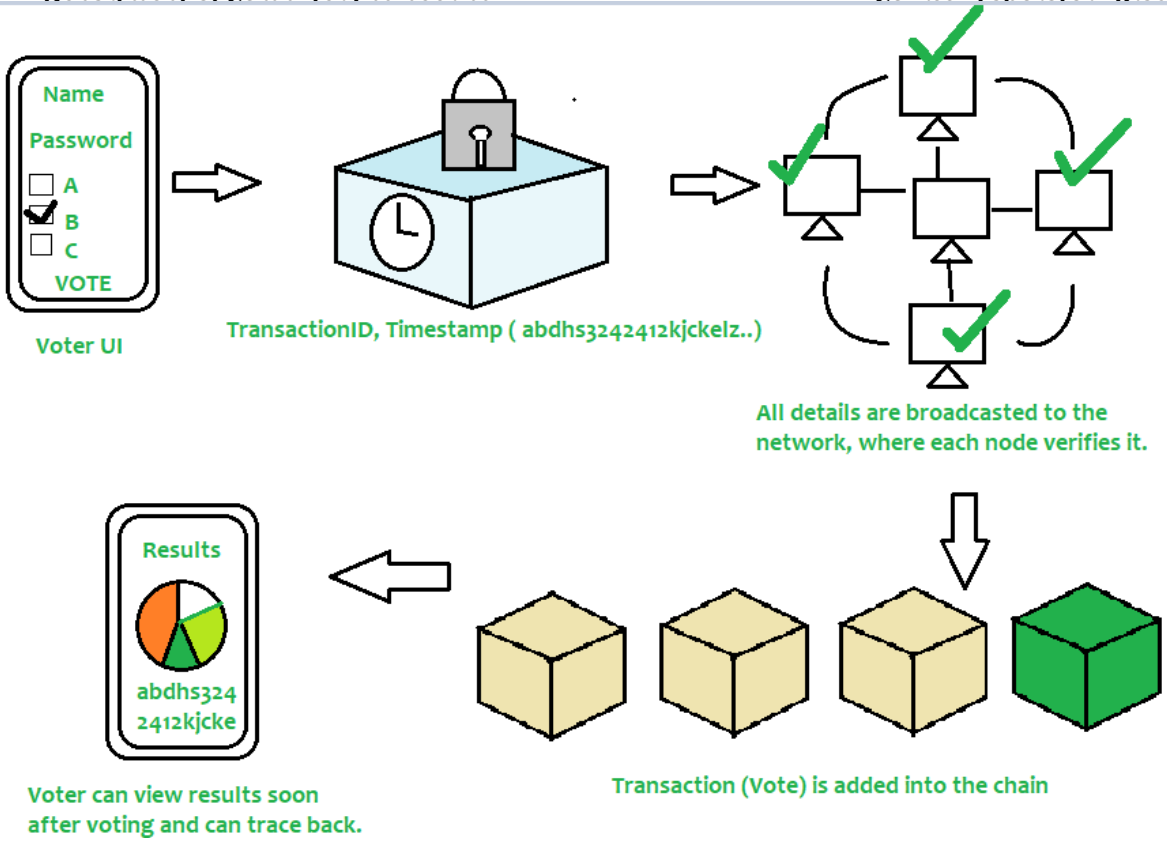
To avoid this, if system is integrated with blockchain- a special property called immutability protects system. Consider SQL, PHP, or any other traditional database systems. You can insert, update, or delete votes. But in a blockchain you can just **insert data** but cannot update or delete. Hence when you insert something, it stays there forever and no one can manipulate it- Thus name immutable ledger.
But Building a blockchain system is not enough. It should be decentralized i.e if one server goes down or something happens on a particular node, other nodes can function normally and do not have to wait for victim node's recovery.

So a gist of advantages are listed below:

- You can vote anytime/anywhere (During Pandemics like COVID-19 where it's impossible to hold elections physically
- Secure
- Immutable
- Faster
- Transparent

Let's visualize process

It is always interesting to learn things if it's visually explained. Hence diagram given below explains how the blockchain voting works.



According to above diagram, voter needs to enter his/her credentials in order to vote. All data is then encrypted and stored as a transaction. This transaction is then broadcasted to every node in network, which in turn is then verified. If network approves transaction, it is stored in a block and added to chain. Note that once a block is added into chain, it stays there forever and can't be updated. Users can now see results and also trace back transaction if they want.

Since current voting systems don't suffice to security needs of modern generation, there is a need to build a system that leverages security, convenience, and trust involved in voting process. Hence voting systems make use of Blockchain technology to add an extra layer of security and encourage people to vote from any time, anywhere without any hassle and makes voting process more cost-effective and time-saving.

Code:

https://github.com/sherwyn11/E-Voting-App

Conclusion

Hence we have successfully made this mini project.