SHALAKA FOUNDATION'S

KEYSTONE SCHOOL OF ENGINEERING

Department of Computer Engineering

LABORATORY MANUAL

LABORATORY PRACTICE III

BLOCKCHAIN TTECHNOLOGY

SEMESTER-I

Subject Code: 410246

TEACHING SCHEME
Lectures:3Hrs/Week
Practical:2Hrs/Week

EXAMINATION SCHEME
Practical:50Marks
TermWork:50Marks

-: **Name of Faculty**:
Prof.Vrushali Wankhede

Group C: Blockchain Technology

Any 5 assignments and 1 Mini project are mandatory.

Sr.No.	Title	Date
1	Installation of MetaMask and study spending Ether per transaction.	
2	Create your own wallet using Metamask for crypto transactions.	
3	Write a smart contract on a test network, for Bank account of a customer for following operations: • Deposit money • Withdraw Money • Show balance	
4	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 values.	
5	Write a survey report on types of Blockchains and its real time use cases.	
6	Write a program to create a Business Network using Hyperledger	
7	Mini Project - Develop a Blockchain based application dApp (de-centralized app) for e voting syste	
8	Mini Project - Develop a Blockchain based application for transparent and genuine charity	
9	Mini Project - Develop a Blockchain based application for health related medical records	
10	Mini Project - Develop a Blockchain based application for mental health	

BLOCKCHAIN LAB MANUAL

Assignment 1

Aim: Installation of Metamask and study spending Ether per transaction.

Theory:

Metamask: MetaMask is a type of Ethereum wallet that bridges the gap between the user interfaces for Ethereum (e.g. Mist browsers, DApps) and the regular web (e.g. Chrome, Firefox, websites).

Its function is to inject a JavaScript library called web3.js into the namespace of each page your browser loads. Web3.js is written by the Ethereum core team.

fter adding MetaMask as an extension in chrome and creating an account, set up your account as follows –

How to Install and Use Metamask

Step 1: Go to Chrome Web Store Extensions Section.

Step 2: Search *MetaMask*.

Step 3: Check the number of downloads to make sure that the legitimate MetaMask is being installed, as hackers might try to make clones of it.

Step 4: Click the *Add to Chrome* button.

Step 5: Once installation is complete this page will be displayed. Click on the *Get Start*.

Fauset using test network:

Step 1: Select *goerli Test Network* from a list of available networks as below

Step 2: Request test ether form https://goerli-faucet.pk910.de/ and using mining you will get the goerlieth

Step 3: Check the test network address will with faucet balance.

Steps to deploy your contract (step 1-7 make doc file of all steps with step description). Ether spending by creating contract text doc and screen shots.

Step 1: Open Remix IDE in your browser. After opening click on + and write the filename as follows:

Screenshot:

Step 2: Write contract to display message.

Screenshot:

Step 3: After compilation and move to deploy section just below the compilation and select Injected Provider – MetaMask in place of Remix VM as shown below –

Screenshot:

Step 4: Now your contract is ready to be deployed. Click on deploy button and the MetaMask will ask for confirmation as follows –

Screenshot:

Step 5: After confirmation, the deployed contract will look like –

Screenshot:

Step 6: Expand the deployed contract as below and get the output using the *get_output() function:*

Screenshot:

Step 7: Now, to verify whether your transaction (process) executed successfully, you can check your balance on MetaMask.

MetaMask comes pre-loaded with fast connections to the Ethereum blockchain and several test network.This allows you to get started without synchronizing a full node,

while still providing the option to upgrade your security and use the blockchain provider of your choice.
Today, MetaMask is compatible with any blockchain that exposes an Ethereum-compatible JSON RPC API(opens new window), including custom and private blockchains.

Ether: Ether is the transactional token that facilitates operations on the Ethereum network. All of the programs and services linked with the Ethereum network require computing power, equipment, internet connections, and maintenance. Ether is the payment users give to network participants for executing their requested operations on the network. Metaphorically speaking, it is more accurate to refer to ether as the "gas" that powers the network. Gas is the term the community uses to refer to the exchange of ether for the work done to verify transactions and secure the blockchain.

OUTPUT:

Screen shots of metamask installation steps.

Ether through goerli test network screen shots.

Etherscan screenshots.

Conclusion: In this assignment we done with metamask and ether spending per transaction.

Assignment 2

Aim: Create your own wallet using Metamask for crypto transactions.

Theory: Create Metamask Wallet.

Metamask: MetaMask is a type of Ethereum wallet that bridges the gap between the user interfaces for Ethereum (e.g. Mist browsers, DApps) and the regular web (e.g. Chrome, Firefox, websites).

Steps to create wallet:

Step 1*: This is the first time creating a wallet, so click the Create a Wallet button. If there is already a wallet then import the already created using the Import Wallet button. ed* button.

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: Click the buttons respective to the order of the words in your seed phrase. In other words, type the seed phrase using the button on the screen. If done correctly the *Confirm* button should turn blue.

Output: All steps screenshot each screen shot should have label.

Conclusion: This is how we perform ether per transaction.

Assignment 3

Aim:
Write a smart contract on a test network, for Bank account of a customer for following operations: Deposit money, Withdraw Money , Show balance.

Theory:
Smart contracts are simply programs stored on a block chain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met. A smart contract is an agreement between two people or entities in the form of computer code programmed to execute automatically

How does a smart contract work:
The operation of a smart contract is similar to other blockchain transfers. These are the necessary steps:
1. A user initiates a transaction from their block chain wallet.
2. The transaction arrives at the distributed database, where the identity is confirmed.
3. The transaction, which may be a transfer of funds, is approved.
4. The transaction includes the code that defines what type of transaction is to be executed.
5. The transactions are added as a block within the block chain.
6. Any change in contract status follows the same process to be updated.

Algorithm:
Flowchart:
Output:
Ether Transaction:

Screenshot:

Debugging:

Screenshot

Ether deduct from one address to another:

Screenshot:

Conclusion:

We create a smart contract on a test network, for Bank account of a customer for following operations: Deposit money ,Withdraw Money, Show balance.

Assignment 4

Aim:
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 values.

Theory:

Structure, Fallback ,Array in solidity. Syntax and working

Defining a Struct
To define a Struct, you must use the struct keyword. The struct keyword defines a new data type, with more than one member. The format of the struct statement is as follows −

```
struct struct_name {
  type1 type_name_1;
  type2 type_name_2;
  type3 type_name_3;
}
```

Example

```
struct Book {
  string title;
  string author;
  uint book_id;
}
```

Declaring Arrays
To declare an array of fixed size in Solidity, the programmer specifies the type of the elements and the number of elements required by an array as follows −

```
type arrayName [ arraySize ];
```

Initializing Arrays
You can initialize Solidity array elements either one by one or using a single statement as follows −

```
uint balance[3] = [1, 2, 3];
```

Creating dynamic memory arrays
Dynamic memory arrays are created using new keyword.

```
uint size = 3;
uint balance[] = new uint[](size);
```

Accessing Array Elements
An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example −

```
uint salary = balance[2];
```

Fallback: The solidity fallback function is executed if none of the other functions match the function identifier or no data was provided with the function call. Only one unnamed function can be assigned to a contract and it is executed whenever the

contract receives plain Ether without any data. To receive Ether and add it to the total balance of the contract, the fallback function must be marked payable. If no such function exists, the contract cannot receive Ether through regular transactions and will throw an exception.
Properties of a fallback function:
Declare with fallback() and have no arguments.
If it is not marked payable, the contract will throw an exception if it receives plain ether without data.
Can not return anything.
Can be defined once per contract.
It is also executed if the caller meant to call a function that is not available or receive() does not exist or msg.data is not empty.
It is mandatory to mark it external.
It is limited to 2300 gas when called by another function by using transfer() or send() method . It is so for as to make this function call as cheap as possible.

Algorithm:
Flowchart:

Fallback Code and Output:

Conclusion:
This is how we create a Student data. Use the following constructs:
- Structures
- Arrays
- Fallback

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

Assignment 5

Aim: Write a survey report on types of Block chains and its real time use cases.

Theory:

Permissionless Blockchain

It is also known as trustless or public blockchains, are available to everyone to participate in the blockchains process that use to validate transactions and data. These are used in the network where high transparency is required.

Characteristics:

- Permissionless blockchain has no central authority.
- The platform is completely open-source.
- Full transparency of the transaction.
- Heavy use of tokens.

Permissioned Blockchain

These are the closed network only a set of groups are allowed to validate transactions or data in a given blockchain network. These are used in the network where high privacy and security are required.

Characteristics:

- A major feature is a transparency based on the objective of the organization.
- Another feature is the lack of anatomy as only a limited number of users are allowed.
- It does not have a central authority.
- Developed by private authority.
- As few nodes are involved performance and scalability are increased.
- Anytime owner and operator can change the rules as per their need.

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.

Explain any real time use case of blockchain in detail i.e online voting system/any gov sector/healthcare

Conclusion: This is how we study real time cases of blockchain.

Assignment 6

Aim: Write a program to create a Business Network using Hyperledger.

Theory: Create Business Network Using Hyperledger Composer

Open Composer Playground (note, this link will take you to the web Composer Playground - you can also follow along in a local version if you've already installed the development environment).

You should see the My Business Networks screen. The My Business Networks page shows you a summary of the business networks you can connect to, and the identities you can use to connect to them. Don't worry about this too much for the time being, as we're going to create our own network.

Creating a new business network

Next, we want to create a new business network from scratch. A business network has a couple of defining properties; a name, and an optional description. You can also choose to base a new business network on an existing template, or import your own template.

1. Click Deploy a new business network under the Web Browser heading to get started.
2. The new business network needs a name, let's call it `tutorial-network`.
3. Optionally, you can enter a description for your business network.
4. Next we must select a business network to base ours on, because we want to build the network from scratch, click empty-business-network.
5. Now that our network is defined, click Deploy.

As you can see, we're in the Define tab right now, this tab is where you create and edit the files that make up a business network definition, before deploying them and testing them using the Test tab.

As we selected an empty business network template, we need to modify the template files provided. The first step is to update the model file. Model files define the assets, participants, transactions, and events in our business network.

For more information on our modeling language, check our documentation.

Click the Model file to view it.

1. Delete the lines of code in the model file and replace it with this:

This domain model defines a single asset type `Commodity` and single participant type `Trader` and a single transaction type `Trade` that is used to modify the owner of a commodity.

Now that the domain model has been defined, we can define the transaction logic for the business network. Composer expresses the logic for a business network using JavaScript functions. These functions are automatically executed when a transaction is submitted for processing.

For more information on writing transaction processor functions, check our documentation.

1. Click the Add a file button.
2. Click the Script file and click Add.
3. Delete the lines of code in the script file and replace it with the following code:

Copy

This function simply changes the `owner` property on a commodity based on the `new Owner` property on an incoming `Trade` transaction. It then persists the modified `Commodity` back into the asset registry, used to store `Commodity` instances.The first thing we should add to our business network is two participants.

1. Ensure that you have the Trader tab selected on the left, and click Create New Participant in the upper right.
2. What you can see is the data structure of a *Trader* participant. We want some easily recognizable data, so delete the code that's there and paste the following:

Now that we have two Trader participants, we need something for them to trade. Creating an asset is very similar to creating a participant. The Commodity we're creating will have an owner property indicating that it belongs to the Trader with the trade Id of `TRADER1`.

1. Click the Commodity tab under Assets and click Create New Asset.
2. Delete the asset data and replace it with the following:

.To test the Trade transaction: Click the Submit Transaction button on the left.Ensure that the transaction type is Trade. Replace the transaction data with the following, or just change the details:

Click Submit.

Conclusion: This is how we create business network on hyper ledger.