BCT

Assignment No- 01.

* Title:- Installation of Metamask and study Spending Ether per Transaction

* Objective: Understand and explore the working of Blockchain Technology and its applications

* Theory:

Metamask:

1) Introduction: Metamask is a plug-in Ethereum crypto wallet for chrome onboard users. Available as a browser extension and as a mobile app, metamask equips us with key vault, secure login & token wallet - everything we need to manage our digital s assests. Metamask provides the simplest yet most secure way to connect to blockchain based applications.

2) Metamask setup:
complete information and study guide about metamask can be found at its official website (metamask.io) we need to choose the right browser (Chrome is recommended) and follow its installation instruction.
When we are creating a new Metamask account, here are some key points we need to pay attention to :

1) Creating a new strong password is extremely important because it encrypts private key.

- Private keys give access to all of our Ether or other tokens. so it is better to have a strong password here.

- Secret backup phrase, which includes 12 mnemonic words, will popup after setting up the password. we need to write this phrase on a piece of paper or store it in secure location because secret & backup phrase makes it easier to backup & restore our account if we logout our account or accidently clear browser history.

We are now able to use & interact with metamask.

3) Steps to create metamask:-

The steps for installation are:

Step1: Search on 'google 'Metamask''

Step2: Download the metamask. Click on the button 'Add to chrome'.

Step3: metamask wallet installation. click on the metamask extension and click 'Get started'

Step4: ~~Creat~~ click on 'create a wallet'.

Step5: Click on 'I agree'.

Step6: Create a password '******' and confirm the password '******'. click on 'I agree' click on 'create'

Step7: 'Metamask' account created successfully.

* Conclusion:

Successfully created Metamask wallet.

Assignment No- 02

* Title: write a smart contract on a test work, for bank account of a customer for foll. oper^ns.
  1) Deposit money
  2) Withdraw money
  3) Show Balance

* Objective: Understand and explore the working of BCT and its applications.

* Theory:

what is a smart contract?

A smart contract is a self-executing contract with the terms of the ~~aggre~~ agreements contained therein exist across a distributed, decentralised blockchain network. The code controls the execution, and transactions are trackable and irreversible.

Anyone ~~to~~ can write a smart contract and deploy it to the network.

Solidity:

Solidity is an object-oriented programming language created specifically by the Ethereum N/w team for constructing and designing smart contracts on Blockchain platforms.

It's actually a statically typed curly-braces programming language that has familiar features that you might recognize from other languages.

solidity's main influences are Javascript ,c++ and Python.

* The Remix Editor:

What is a remix platform?

Remix is a powerful , open source tool that helps you write solidity contracts straight from the browser. Remix IDE is used for the entire journey of smart contract developers at every knowledge level.
The IDE comes in 2 flavors (web app or desktop app) and as a vscode Extension.

steps to develop an Ethereum smart contract:

step1: create a wallet at metamask.
step2: Select any one test network.
Step3: Add some dummy Ethers to your wallet.
Step4: Use editor remix to write the smart contract in solidity.
step5: create a .sol extension file
step6: A sample smart contract code to create ERC20 tokens.
Step7: Deploy your contract.

Banking smart contract:

This smart contract will have all basic functionalities like:-
1) Account creation
2) Deposit Amount
3) Withdraw Amount
4) Transfer Amount
5) send Amount to wallet.

First add solidity compiler version

pragma solidity >= 0.4.22 < 0.9.0

Then creating Banking contract:
contract banking {....}

create variables:
mapping (address => uint) public userAccount;
mapping (address => bool) public userExists;

Now create f^ns for each operations.

1) createAcc() function
2) deposit() function
3) withdraw (uint amount) function
4) TransferAmount ( ) function
5) sendAmount( ) function

* conclusion:

In this way we studied what is smart contract and how to write and deploy it.

BCT

Assignment No-03

* Title: write a program in solidity to create student data. use the following constructs:
1) structures 2) Arrays 3) Fallback.
Deploy this as smart contract on ~~ant~~ ethereum & observe the transaction fee and Gas values.

* Objective: Understand and explore the working of Blockchain Technology & its applications.

* Theory:
Solidity is an object-oriented, high-level language for implementing smart contracts.

Following are the constructs of smart solidity:

1) Structures:

struct:

Structs in solidity allows you to create more complicated data types that have multiple properties. You can define your own type by creating a struct.
structs can be declared outside of a contract & imported in another contract. It's used to represent a record.
struct keyword is used to create/define a structure

Syntax:

```
struct <structure_name> {
  <datatype> variable_1;
  <datatype> variable_2;
}
```

2) Arrays:

- Arrays are data structures that store the fixed collection of elements of the same datatypes in which each & every element has a specific loc^{n} called index.
- Instead of creating individual variables of same type, we create one array of required size & access the elements in it using index.
- In solidity array can be fixed size or dynamic size.

Creating an array:

Syntax: <datatype><array name>[size] = <initialization>

Fixed size arrays:

size of array should be predefined. The total no. of elements should not exceed size of array.

Dynamic arrays:

The size of array is not predefined when it is declared. As the elements are added size of array changes & at runtime, the size of the array will be determined.

* Array operations:

1) Accessing array elements: done using index which starts from 0
2) length of array: used to check no of elements present in the array.
3) Push: used when new element is to be added in dynamic array. New element is always added at the last position.
4) Pop: used to remove last element from any dynamic array.

8) Fall back:

To solidity fallback is executed if none of the other f^n's match the f^n identifier or no data was provided with the f^n call. To receive Ether & add it to the total balance of the contract, the fallback f^n must be marked payable.

Properties of a fallback f^n:-

1) Has no name or arguments
2) If it is not marked payable, the contract will throw an exception if it receives plain ether without data.
3) Can not return anything
4) Can be defined once per contract
5) It is also calculated if the caller meant to call a f^n that is not avoidable.
6) It is mandatory to make it external.
7) It is limited to 2300 gas when called by another function It is so far as to make this function call as cheap as possible.

- only one unamed function can be assigned to a contract and is it is executed whenever the contract receives plain Ether without any data. If no such function exists, the contract cannot receive Ether through regular transactions & will throw an exception.

Conclusion: In this way we studied what is smart contract and how to create smart contract for students data using different constructs.

BCT

Assignment No-04.

* Write a survey Report on types of Blockchains and its real time use cases.

* Content/Theory:
1] Introduction to Blockchain
2] Overview of Blockchain history
3] Types of Blockchain with real time use cases of each.
4] Applications of Blockchain
Conclusion

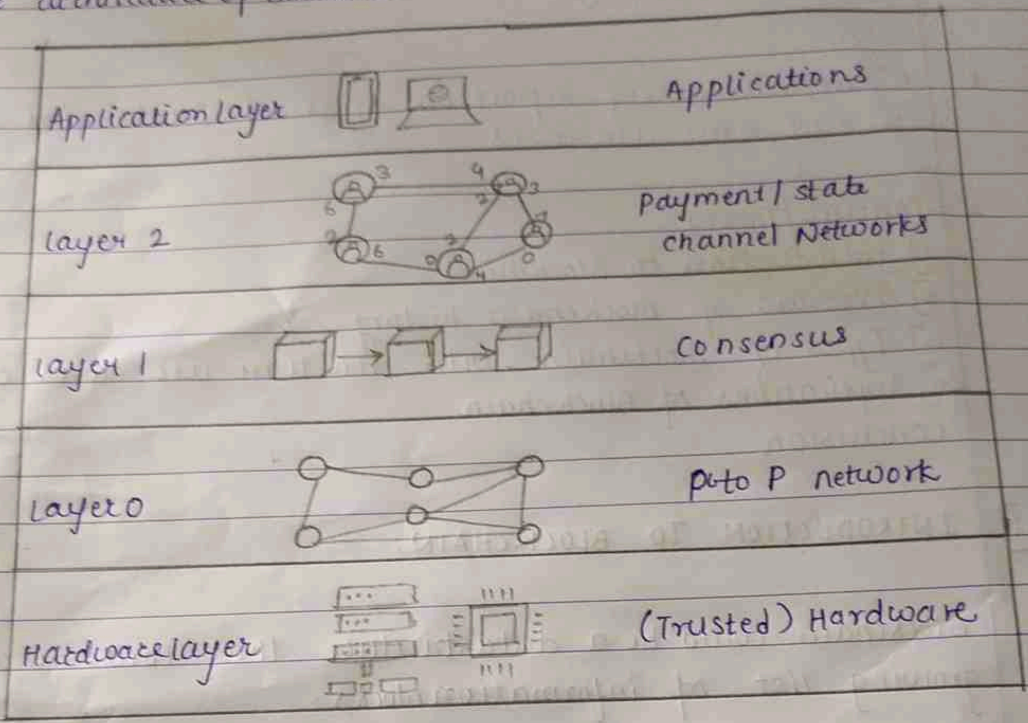
*1] INTRODUCTION TO BLOCKCHAIN:

Blockchain could be a datastructure that could be a growing list of information blocks.

What is Blockchain ?
The blockchain is a distributed database of records of all transactions or digital events that have been executed and shared among participating parties. Each transaction is verified by the majority of participants of the system.

Bitcoin is the most popular cryptocurrency an example of the Blockchain.
Anything of value like land Assests, cars etc can be recorded on Blockchain as a Transaction.

The architecture of Blockchain:-



Application layer	Applications
layer 2	Payment / state channel Networks
layer 1	Consensus
Layer 0	P-to P network
Hardwarelayer	(Trusted) Hardware

2] OVERVIEW OF BLOCKCHAIN HISTORY:

Blockchain has the potential to grow to be a bedrock of the world wide record-keeping systems, but was launched just 10 years ago. It was created by the unknown persons behind the online cash currency bitcoin, under the pseudonym of Satoshi Nakamoto.

Brief History of Blockchain:

- 1991 - A cryptographically secured chain of blocks is described for the first time by Stuaet Haber & W scott Stornetta.

- 1998:- computer scientist Nick Szabo works on 'bit gold', a decentralised digital currency.

- 2000:- stefan konst published his theory of cryptographic secured chains, plus ideas for implementation.

- 2008: Developer(s) working under the pseudonym Satoshi Nakamoto released a white paper establishing the model for a blockchain

- 2009: Nakamoto implemented the first blockchain as the public ledger for transactions made using bitcoin

- 2014: Blockchain Technology is separated from the currency and its potential for the finanancial, interorganisational transaction is explored. Blockchain 2.0 is born, referring to applications beyond currency.

3] TYPES OF BLOCK CHAIN WITH REALTIME USE CASES OF EACH:

There are basically 4 types of blockchains

1) Permissionless (Public)
2) ~~Pri~~ Permissioned (Private)
3) Hybrid
4) consortium.

1] Permissionless (Public Block chain):-

- It is for the public & of the public.
- There is no one in charge, & anyone can take part

in the process.

- These blockchains are open & transparent.
- Decisions are made through consensus mechanism.

& Use cases:

- financial services can write smart contracts b/w customers & their banking institutions.
- Healthcase can use it to write smart contracts between insurers & hospitals as well as patients & hospitals.

eg: Bitcoin, Ethereum, Litecoin.

2] Permissioned (Private) Blockchain:-

- It is like a private assest of the individual or an organisation.
- Private blockchain has a in-charge who monitors imp. task & give access to read or block access.
- They are internal to the company so companies will not want it accessible ~~to~~by the public.

& Use cases:

- Used for cross-border payments, trade finance, & settlement systems.
- Eg: Hyperledger, Ethereum Enterprise, R3 corda, Ripple

3] Hybrid Blockchain:

combines elements of both public & private blockchains.

- Use cases:

· Enterprise services

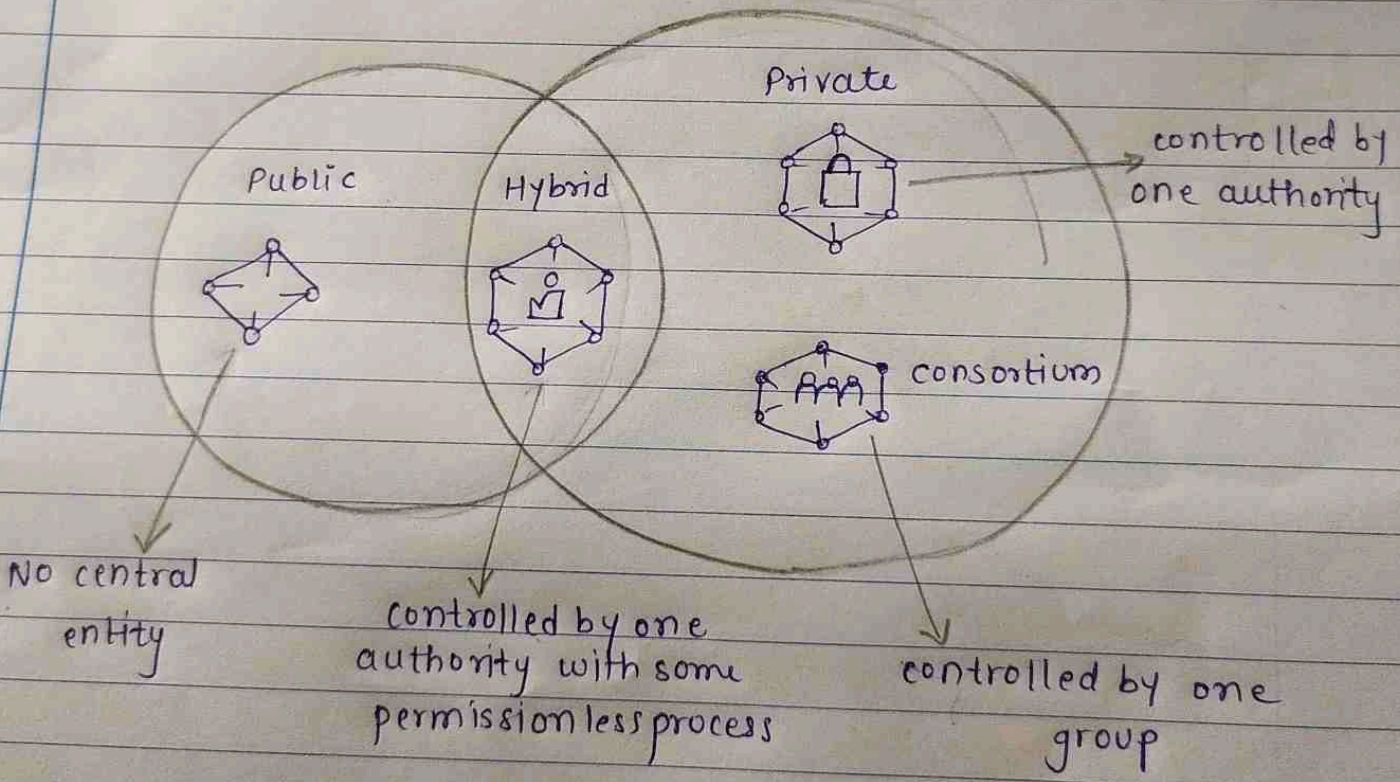
· Hybrid IOT

- Global Trade & finance
- Banking
- Government.

4] Consortium Blockchain:

- Selected members can make transactions & review/audit blockchain.
- Neither private nor public.
- Use cases:
  R3 corda is used in finance sector for secure & efficient interbank transactions.

Overview of Types of Block chains:


Private
controlled by one authority
Public
Hybrid
consortium
No central entity
controlled by one authority with some permissionless process
controlled by one group

* Applications of Blockchain:

1) Asset management:- Involves handling of & exchange of different ass