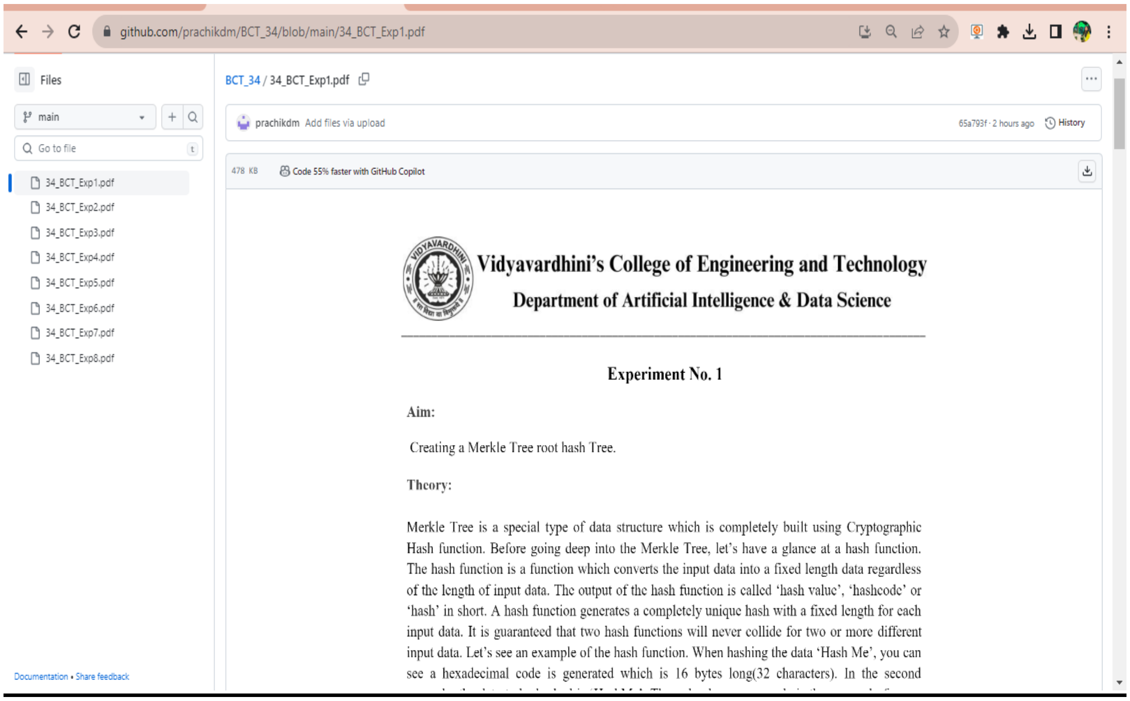
Experiment No: 01

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp1.pdf

Snapshot:



Vidyavardhini’s College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 1

Aim:

Creating a Merkle Tree root hash Tree.

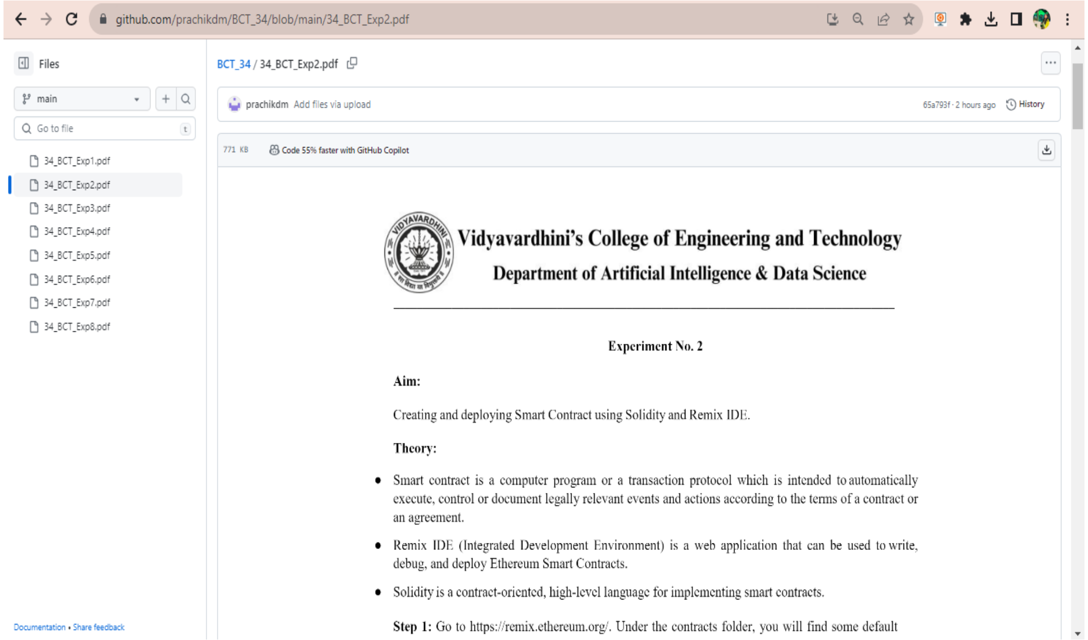
Theory:

Merkle Tree is a special type of data structure which is completely built using Cryptographic Hash function. Before going deep into the Merkle Tree, let’s have a glance at a hash function. The hash function is a function which converts the input data into a fixed length data regardless of the length of input data. The output of the hash function is called ‘hash value’, ‘hashcode’ or ‘hash’ in short. A hash function generates a completely unique hash with a fixed length for each input data. It is guaranteed that two hash functions will never collide for two or more different input data. Let’s see an example of the hash function. When hashing the data ‘Hash Me’, you can see a hexadecimal code is generated which is 16 bytes long(32 characters). In the second

Experiment No. 02

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp2.pdf

Snapshot:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 2

Aim:

Creating and deploying Smart Contract using Solidity and Remix IDE.

Theory:

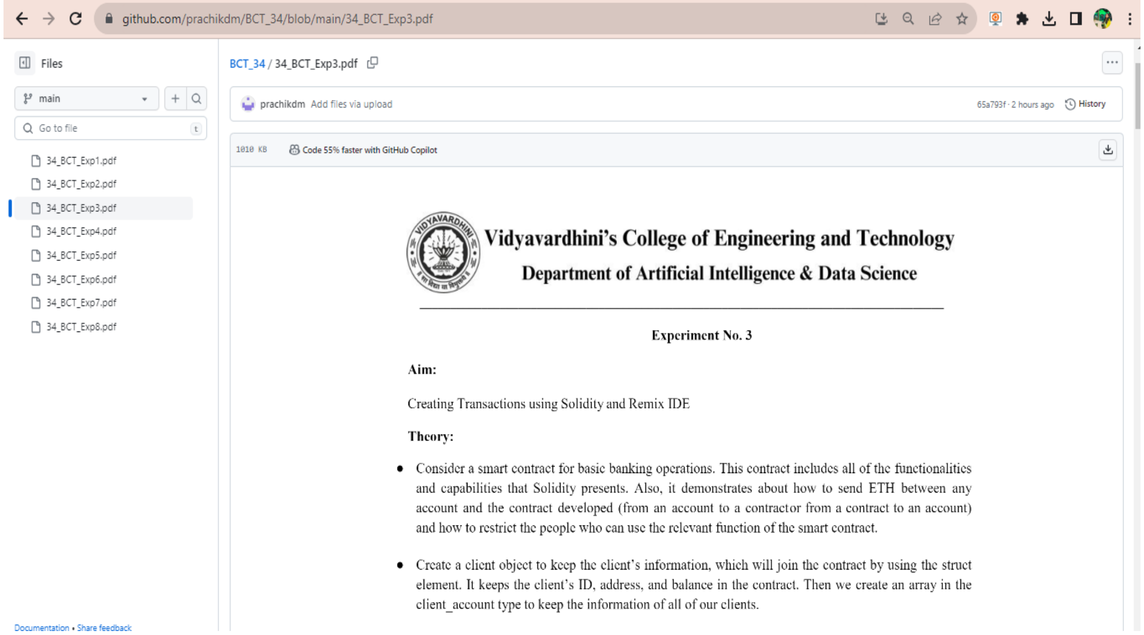
- Smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement.
- Remix IDE (Integrated Development Environment) is a web application that can be used to write, debug, and deploy Ethereum Smart Contracts.
- Solidity is a contract-oriented, high-level language for implementing smart contracts.

Step 1: Go to https://remix.ethereum.org/. Under the contracts folder, you will find some default

Experiment No: 03

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp3.pdf

Snapshot:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 3

Aim:

Creating Transactions using Solidity and Remix IDE

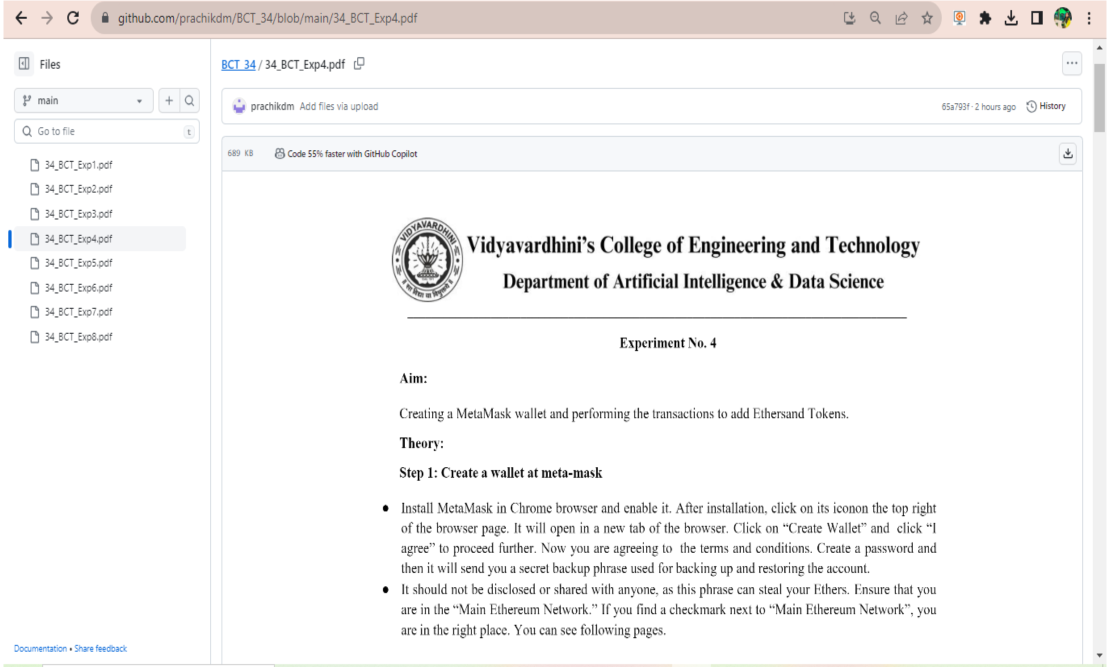
Theory:

- Consider a smart contract for basic banking operations. This contract includes all of the functionalities and capabilities that Solidity presents. Also, it demonstrates about how to send ETH between any account and the contract developed (from an account to a contractor from a contract to an account) and how to restrict the people who can use the relevant function of the smart contract.
- Create a client object to keep the client's information, which will join the contract by using the struct element. It keeps the client's ID, address, and balance in the contract. Then we create an array in the client_account type to keep the information of all of our clients.

Experiment No: 04

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp4.pdf

Snapshot:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 4

Aim:

Creating a MetaMask wallet and performing the transactions to add Ethersand Tokens.

Theory:

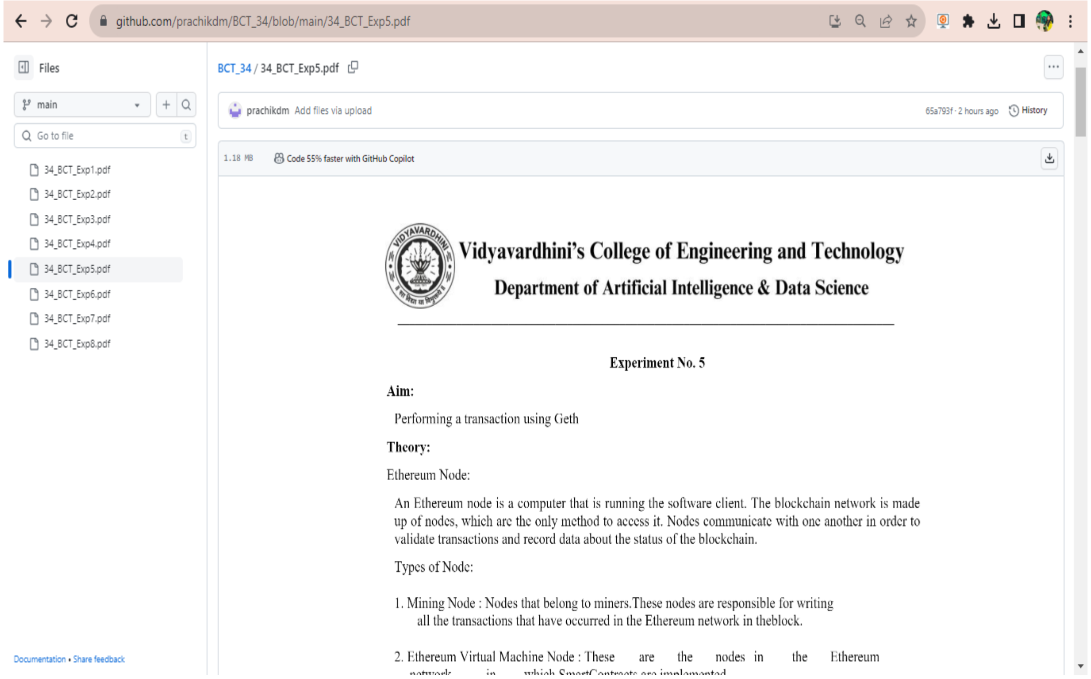
Step 1: Create a wallet at meta-mask

- Install MetaMask in Chrome browser and enable it. After installation, click on its iconon the top right of the browser page. It will open in a new tab of the browser. Click on "Create Wallet" and click "I agree" to proceed further. Now you are agreeing to the terms and conditions. Create a password and then it will send you a secret backup phrase used for backing up and restoring the account.
- It should not be disclosed or shared with anyone, as this phrase can steal your Ethers. Ensure that you are in the "Main Ethereum Network." If you find a checkmark next to "Main Ethereum Network", you are in the right place. You can see following pages.

Experiment No: 05

Github Url: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp5.pdf

Snapshot:



Vidyavardhini’s College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 5

Aim:

Performing a transaction using Geth

Theory:

Ethereum Node:

An Ethereum node is a computer that is running the software client. The blockchain network is made up of nodes, which are the only method to access it. Nodes communicate with one another in order to validate transactions and record data about the status of the blockchain.

Types of Node:

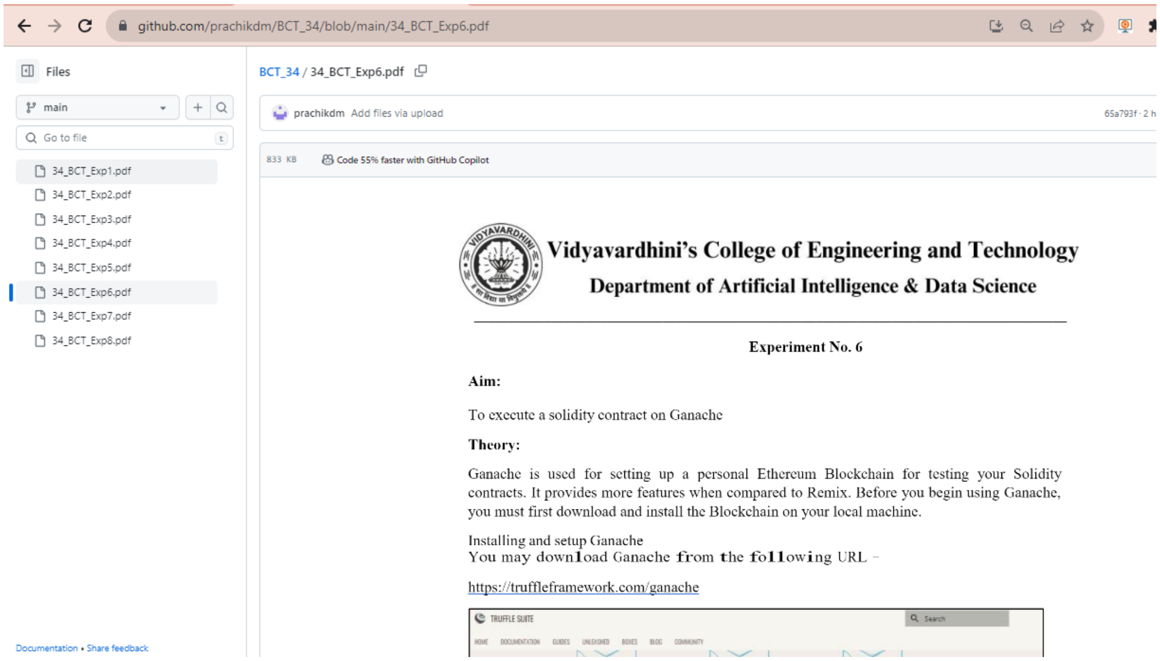
1. Mining Node : Nodes that belong to miners.These nodes are responsible for writing all the transactions that have occurred in the Ethereum network in theblock.

2. Ethereum Virtual Machine Node : These are the nodes in the Ethereum network in which SmartContracts are implemented

Experiment No: 06

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp6.pdf

Snapshot:


github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp6.pdf
Files
main
Go to file
34_BCT_Exp1.pdf
34_BCT_Exp2.pdf
34_BCT_Exp3.pdf
34_BCT_Exp4.pdf
34_BCT_Exp5.pdf
34_BCT_Exp6.pdf
34_BCT_Exp7.pdf
34_BCT_Exp8.pdf
Documentation • Share feedback
BCT_34 / 34_BCT_Exp6.pdf
prachikdm Add files via upload
65a793f · 2 h
833 KB
Code 55% faster with GitHub Copilot
TRUFFLE SUITE
Search
HOME DOCUMENTATION GUIDES UNLEASHED BOXES BLOG COMMUNITY

Vidyavardhini’s College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 6

Aim:

To execute a solidity contract on Ganache

Theory:

Ganache is used for setting up a personal Ethereum Blockchain for testing your Solidity contracts. It provides more features when compared to Remix. Before you begin using Ganache, you must first download and install the Blockchain on your local machine.

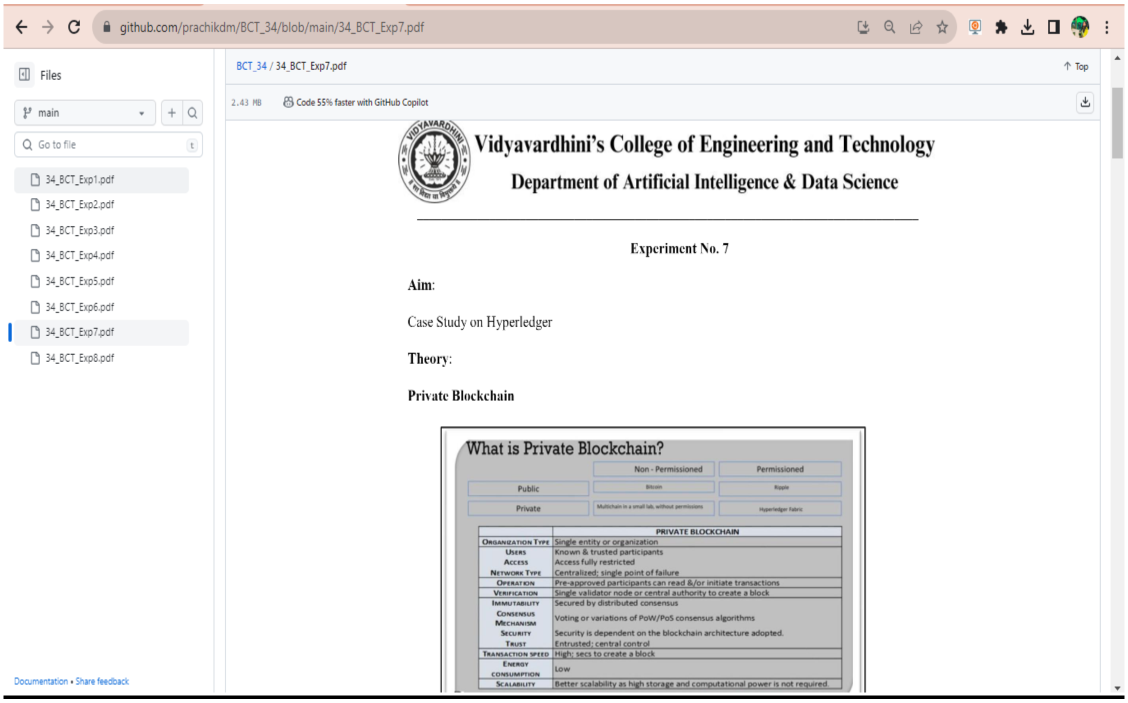
Installing and setup Ganache
You may download Ganache from the following URL –

https://truffleframework.com/ganache

Experiment No: 07

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp7.pdf

Snapshot:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 7

Aim:

Case Study on Hyperledger

Theory:

Private Blockchain

What is Private Blockchain?

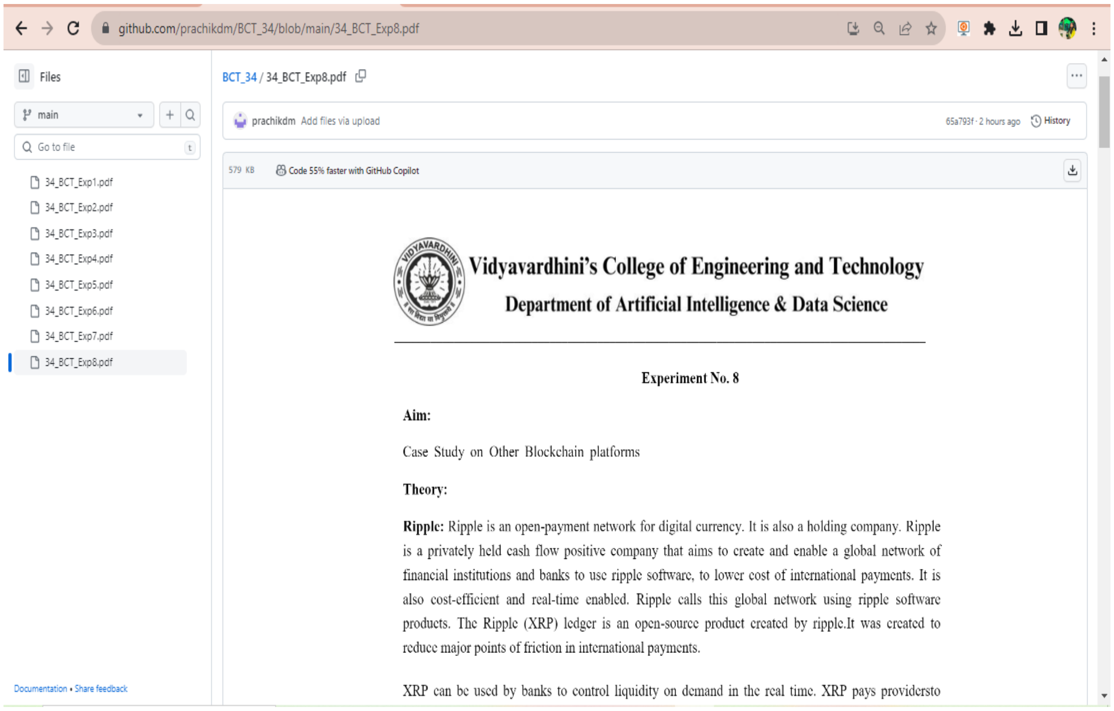
	Non - Permissioned	Permissioned
Public	Bitcoin	Ripple
Private	Multichain in a small lab, without permissions	Hyperledger Fabric

	PRIVATE BLOCKCHAIN
Organization Type	Single entity or organization
Users	Known & trusted participants
Access	Access fully restricted
Network Type	Centralized; single point of failure
Operation	Pre-approved participants can read &/or initiate transactions
Verification	Single validator node or central authority to create a block
Immutability	Secured by distributed consensus
Consensus Mechanism	Voting or variations of PoW/PoS consensus algorithms
Security	Security is dependent on the blockchain architecture adopted.
Trust	Entrusted; central control
Transaction speed	High; secs to create a block
Energy consumption	Low
Scalability	Better scalability as high storage and computational power is not required.

Experiment No: 08

GitHub URL: https://github.com/prachikdm/BCT_34/blob/main/34_BCT_Exp8.pdf

Snapshot:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 8

Aim:

Case Study on Other Blockchain platforms

Theory:

Ripple: Ripple is an open-payment network for digital currency. It is also a holding company. Ripple is a privately held cash flow positive company that aims to create and enable a global network of financial institutions and banks to use ripple software, to lower cost of international payments. It is also cost-efficient and real-time enabled. Ripple calls this global network using ripple software products. The Ripple (XRP) ledger is an open-source product created by ripple.It was created to reduce major points of friction in international payments.

XRP can be used by banks to control liquidity on demand in the real time. XRP pays providersto