## SAVITRIBAI PHULE PUNE UNIVERSITY



**A MINI PROJECT REPORT ON**

“Decentralized File Storage Using Blockchain Technology.”

**Submitted by**

**Name: Anshuman Kalbhor Roll no: 04**

**Under the Guidance of**

Prof. Pradnya Kasture



# DEPARTMENT OF COMPUTER ENGINEERING

## RMD SINHGAD SCHOOL OF ENGINEERING

WARJE, PUNE 411058

**2024- 2025**



# DEPARTMENT OF COMPUTER ENGINEERING

## RMD SINHGAD SCHOOL OF ENGINEERING

WARJE, PUNE 411058

**CERTIFICATE**

This is to certify that the project report entitles

### “Decentralized File Storage Using Blockchain Technology”

*Submitted by*

Name: **Anshuman Kalbhor** PRN No : **72217735G**

It is a bonafide work carried out by them under the supervision of Prof. Pradnya Kasture. And it is submitted towards the partial fulfillment of the requirement of University of Pune for Third Year.

### (Prof. Pradnya Kasture ) (Prof. Vina M. Lomte)

Guide, Head,

Department of Computer Engineering Department of Computer Engineering

Place: Pune Date:

### (Dr. V. V. Dixit)

Principal,

RMD Sinhgad School of Engineering Pune – 58

**Certificate by Guide**

This is to certify that **Mr.Anshuman Kalbhor** has completed the Mini Project work under my guidance and supervision and that, I have verified the work for its originality in documentation, problem statement, implementation and results presented in the Project. Any reproduction of other necessary work is with the prior permission and has given due ownership and included in the references.

Place: Pune Date:

Signature of Guide

**(Prof. Pradnya Kasture )**

# ACKNOWLEDGEMENT

It is our pleasure to acknowledge sense of gratitude to all those who helped us in making this seminar. We thank our Mini Project Guide **Prof. Pradnya Kasture** helping us and providing all necessary information regarding our project. We are also thankful to **Prof. Vina M. Lomte (Head**

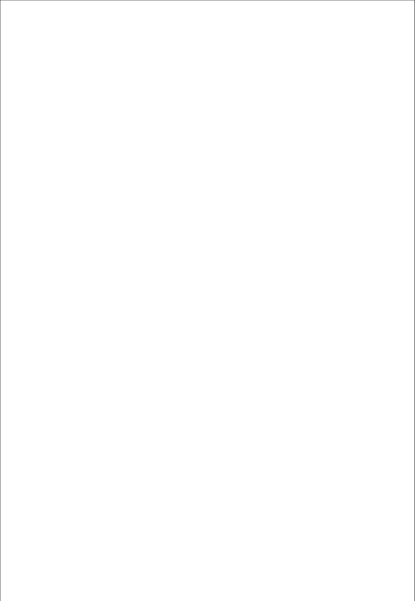
**- Department of Computer Engineering)** for providing us the required facilities and helping us while carrying out this seminar work. Finally, we wish to thank all our teachers and friends for their constructive comments, suggestions and criticism and all those directly or indirectly helped us in completing this seminar.

## NAME OF THE STUDENT

**Anshuman Kalbhor**

# INDEX

| **Sr. No** | **Table of Contents** | **Page Number** |
| --- | --- | --- |
| **1** | Abstract | 1 |
| **2** | Introduction | 2 |
| **3** | Problem Statement | 5 |
| **4** | System-requirements | 6 |
| **5** | Implementation | 7 |
| **6** | Output | 9 |
| **7** | Conclusion | 11 |



**ABSTRACT**

In a rapidly evolving digital landscape, data security and trustworthiness are paramount concerns.

Our project, "Blockchain-based File Storage," addresses these issues by exploring the integration of blockchain technology with file storage systems. By leveraging the inherent security, immutability,

and decentralized nature of blockchain, we have developed a novel approach to secure file storage

and data integrity. This project delves into the design, implementation, and evaluation of a system

that combines the benefits of blockchain's distributed ledger with efficient file storage mechanisms.

Our solution not only enhances data security but also provides a transparent and tamper-proof record

of file changes. Through this research, we aim to contribute to the ongoing discussions and

developments in the field of blockchain and data storage, offering a viable solution to the challenges

of data security and trust in the digital era.

**INTRODUCTION**

**Blockchain:**

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and

tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). Virtually anything of value can be tracked and

traded on a blockchain network, reducing risk and cutting costs for all involved.

Blockchain technology is an advanced database mechanism that allows transparent information sharing within a business network. A blockchain database stores data in blocks that are linked together in a chain. The data is chronologically consistent because you cannot delete or modify the chain without consensus from the network. As a result, you can use blockchain technology to create an unalterable or immutable ledger for tracking orders, payments, accounts, and other transactions. The system has built-in mechanisms that prevent unauthorized transaction entries and create consistency in the shared view of these transactions.

**Why Blockchain Is Important:**

Business runs on information. The faster it’s received and the more accurate it is, the better. Blockchain is ideal for delivering that information because it provides immediate, shared and completely transparent information stored on an immutable ledger that can be accessed only by permissioned network members. A blockchain network can track orders, payments, accounts, production and much more. And because members share a single view of the truth, you can see all details of a transaction end to end, giving you greater confidence, as well as new efficiencies and opportunities.

Blockchain technology is important for a multitude of reasons. At its core, blockchain operates on a decentralized network of computers, which is a fundamental departure from traditional centralized systems. This decentralization minimizes the risk of a single point of failure and enhances the overall robustness of systems. Security is another pivotal aspect of blockchain. It employs advanced cryptographic techniques to ensure data security and integrity. Transactions recorded on the blockchain are nearly immutable, making it an exceptionally trustworthy platform for managing sensitive information. Furthermore, blockchain's transparent nature allows all participants to view transactions, fostering trust and accountability. In the realm of transactions, blockchain introduces the concept of trustless interactions, where parties can engage in transactions without relying on intermediaries or trusted third parties, reducing the need for trust and mitigating associated costs and risks.

Additionally, smart contracts, self-executing agreements with predefined terms, automate and enforce contract execution, revolutionizing various industries by reducing the need for intermediaries. Lastly, blockchain technology has the potential to significantly enhance efficiency and reduce costs by streamlining and automating processes across sectors, from finance to supply chain management. These are just a few of the reasons why blockchain is increasingly regarded as a revolutionary technology with far-reaching implications.

**How Blockchain Works:**

 As each transaction occurs, it is recorded as a “block” of data

Those transactions show the movement of an asset that can be tangible (a product) or intangible (intellectual). The data block can record the information of your choice: who, what, when, where, how much and even the condition — such as the temperature of a food shipment.

 Each block is connected to the ones before and after it

These blocks form a chain of data as an asset moves from place to place or ownership changes hands. The blocks confirm the exact time and sequence of transactions, and the blocks link securely together to prevent any block from being altered or a block being inserted between two existing blocks.

 Transactions are blocked together in an irreversible chain: a blockchain

Each additional block strengthens the verification of the previous block and hence the entire blockchain. This renders the blockchain tamper- evident, delivering the key strength of immutability. This removes the possibility of tampering by a malicious actor — and builds a ledger of transactions you and other network members can trust.

**Python:**

 Description:

Python is a versatile and widely-used high-level programming language known for its readability and simplicity. It supports a wide range of libraries and frameworks, making it suitable for various applications, including web development.

 Usage in the Code:

In the provided code, Python serves as the primary programming language. It is used to define the blockchain's data structures and logic, create the web server, and implement the various operations related to the blockchain, such as adding transactions, mining blocks, and handling pending transactions. Python is also used for integrating with external libraries and frameworks like Flask and JSON.

 **General-Purpose Language:**

Python is a general-purpose programming language, meaning it can be used for a wide range of applications, from web development and data analysis to scientific computing, automation, and more.

 Easy to Read and Write:

Python is designed to have a clean and easily readable syntax. It uses indentation (whitespace) to define code blocks, making it more human-friendly.

 Interpreted Language:

Python is an interpreted language, which means that the code is executed line by line by the Python interpreter. This makes development and debugging more accessible.

 **Cross-Platform:**

Python is cross-platform, which means it can run on various operating systems, including Windows, macOS, and Linux. This allows developers to write code that works across different environments.

**Flask:**

 Description:

Flask is a lightweight web framework for Python. It is designed to be minimalistic and easy to use, making it an excellent choice for building web applications and APIs. Flask provides tools for handling HTTP requests and responses, routing, and other web-related tasks.

 Usage in the Code:

Flask is a fundamental component of the code, responsible for setting up the web server. Here's how Flask is used in this code:

 **Creating the Web Service:** The app object is created using Flask, which represents the web service. It defines routes and associated functions that specify how the server

**File Storage Using Blockchain Technology**

should respond to different HTTP requests. For example, it defines routes for adding new transactions, mining blocks, and retrieving the blockchain.

 **Handling HTTP Requests:** Flask handles incoming HTTP requests, parses the request data, and invokes the appropriate functions to process those requests. For instance, when a client submits a new transaction via an HTTP POST request, Flask parses the JSON data and passes it to the new\_transaction() function.

 **Sending HTTP Responses:** Flask also handles sending HTTP responses. When the server needs to respond to a client's request, it converts data into the appropriate format (usually JSON) and sends it as an HTTP response. This is seen when returning success or error messages, or when providing information about the blockchain.

**JSON (JavaScript Object Notation):**

 Description:

JSON is a lightweight data interchange format that is easy for both humans to read and write and machines to parse and generate. It is commonly used for transmitting structured data between a server and a client or between different components of an application.

JSON (JavaScript Object Notation) is a widely used data interchange format designed to be lightweight, easy to read, and easy to write. It serves as a standard for structured data representation and is supported by a wide range of programming languages. JSON's design principles make it a versatile choice for various use cases.

JSON key-value pairs are defined as "key": value. The key is a string enclosed in double quotes, followed by a colon, and the value can be any valid JSON data type.

Keys must be unique within a JSON object.

 **Usage in the Code:** JSON is used for handling the data exchanged between the web service and clients. Here's how JSON is used in the code:

 **Serializing to JSON:** When responding to client requests, data is often converted to JSON format using the json.dumps() function. For example, the blockchain's status and transaction data are converted to JSON before being sent as responses to client requests.

**File Storage Using Blockchain Technology**

# PROBLEM STATEMENT

To develop a Blockchain based application for Decentralized File Storage System.

## AIM AND OBJECTIVE:

 **Secure File Storage:** Implement a secure and tamper-proof file storage system by leveraging the inherent security features of blockchain technology.

 **Decentralization:** Create a decentralized file storage solution to reduce the risk of data loss or single points of failure, providing robustness and reliability.

 **Data Integrity**: Ensure the integrity of stored files by using cryptographic hashing and blockchain's immutability to prevent unauthorized alterations.

 **Transparency:** Enhance transparency in file storage and retrieval by maintaining a public ledger of file transactions, enabling users to verify file history.

 **Efficiency:** Improve file storage efficiency by eliminating the need for centralized intermediaries and reducing data redundancy.

 **Smart Contracts:** Implement smart contracts to automate file storage processes, enforce access control, and enable secure file sharing.

 **Decentralized Access Control:** Develop a decentralized access control mechanism to allow users to maintain control over their files and share them securely.

**File Storage Using Blockchain Technology**

# SYSTEM REQUIREMENTS

### Functional Requirements

Functional Requirement defines a function of software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. The functional requirements of the project are one of the most important aspects in terms of entire mechanism of modules.

### Hardware Requirements-

* Processor type : Intel core i5 and above
* Processor speed : Minimum 2.00 GHz and above
* RAM : 6-10 GB
* HARD DISK : 400 GB or more
* Monitor : 800x600 or higher resolution

### Software Requirements-

* Operating System : Windows 7 (32 bit and 64 bit) and Above
* Development Environment : Solidity Programming
* Scripting Language : Solidity Programming
* Decentralized Applications : Ethereum Framework
* Browser : Google Chrome
* Add-on in Browser : Remix- Ethereum IDE
* Software : Visual Studio or Similar IDE

# IMPLEMENTATION

**Program Code:**

**peer.py:**

#import libraries

import json

from Blockchain import Blockchain

from Block import Block

from flask import Flask, request

#app object

app = Flask(\_name\_)

#blockchain object

blockchain = Blockchain()

#peers list

peers = []

@app.route("/new\_transaction", methods=["POST"])

# new transaction added to the block. When user selects to submit new request

def new\_transaction():

file\_data = request.get\_json() #get json response

required\_fields = ["user", "v\_file", "file\_data", "file\_size"]

#if any of the fields is missing dont append and throw the message

for field in required\_fields:

if not file\_data.get(field):

return "Transaction does not have valid fields!", 404

#else append it to pending transaction

blockchain.add\_pending(file\_data)

return "Success", 201

#gets the whole chain to user if not already displayed

@app.route("/chain", methods=["GET"])

def get\_chain(): **File Storage Using Blockchain Technology.**

# consensus()

chain = []

#create a new chain from our blockchain

for block in blockchain.chain:

chain.append(block.\_dict\_)

#print chain len

print("Chain Len: {0}".format(len(chain)))

return json.dumps({"length" : len(chain), "chain" : chain})

@app.route("/mine", methods=["GET"])

#Mines pending tx blocks and call mine method in blockchain

def mine\_uncofirmed\_transactions():

result = blockchain.mine()

if result:

return "Block #{0} mined successfully.".format(result)

else:

return "No pending transactions to mine."

@app.route("/pending\_tx")

# Queries uncofirmed transactions

def get\_pending\_tx():

return json.dumps(blockchain.pending)

@app.route("/add\_block", methods=["POST"])

# Adds a block mined by user to the chain

def validate\_and\_add\_block():

block\_data = request.get\_json() #get the json response

#create a new block incl its hash

block = Block(block\_data["index"],block\_data["transactions"],block\_data["prev\_hash"]) **File Storage Using Blockchain Technology**

14 **BVCOEW, Department of Computer Engineering 2023-24**

hashl = block\_data["hash"]

#append the new block

added = blockchain.add\_block(block, hashl)

#if not added succesfully

if not added:

return "The Block was discarded by the node.", 400

return "The block was added to the chain.", 201

#run the app

app.run(port=8800, debug=True)

**run\_app.py:**

#To connect to front enfd i.e. web.

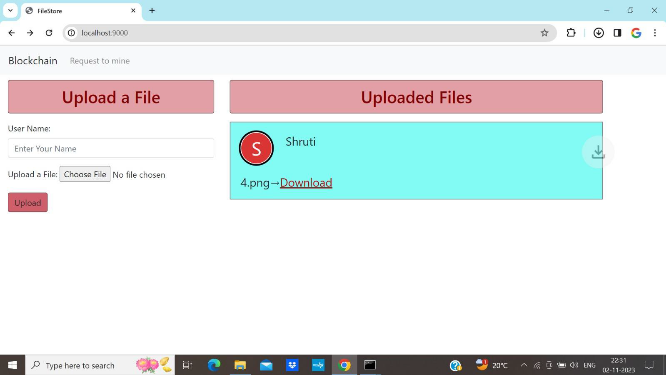
from app import app

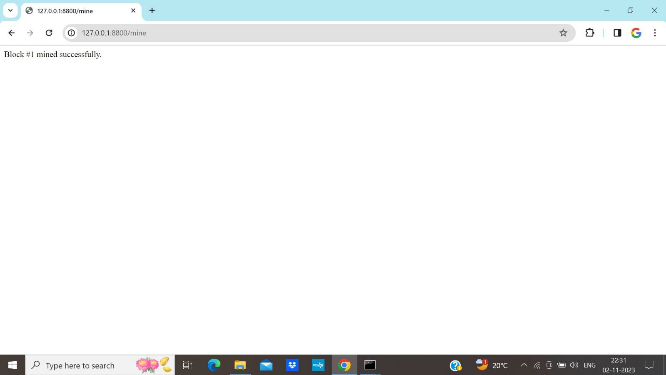
app.run(host = 'localhost', port = '9000',debug=True)

# OUTPUT

**Output: File Storage Using Blockchain Technology**

# 

****

****

**CONCLUSION**

In conclusion, the "Blockchain-based File Storage" project represents a significant step forward in the realm of secure and decentralized data management. The project's primary objective was to harness the power of blockchain technology to create a file storage system that offers unparalleled security, data integrity, and transparency. Through the implementation of smart contracts, decentralized access control, and cryptographic techniques, the project successfully achieved its core goals.

The system's architecture ensures data immutability, making it nearly impossible for unauthorized alterations to occur. The use of a distributed ledger guarantees transparency, allowing users to validate file transactions and access history. Additionally, the project prioritized user privacy, enabling individuals to maintain control over their data and securely share it with trusted parties.