

# **BLOCKCHAIN BASED DECENTRALIZED E-COMMERCE USING ETHEREUM AND SMART CONTRACTS**

## **A PROJECT REPORT**

*Submitted by*

<b>S.SANTHOSH KANNA</b>	<b>142219205079</b>
<b>R.SARANKUMAR</b>	<b>142219205083</b>
<b>T.SURYA</b>	<b>142219205098</b>
<b>V.VIGNESHWARAN</b>	<b>142219205106</b>

*In partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**



**SRM VALLIAMMAI ENGINEERING COLLEGE**

**(AN AUTONOMOUS INSTITUTION)**

**SRM NAGAR, KATTANKULATHUR, CHENGALPATTU**

**ANNA UNIVERSITY :: CHENNAI 600 025**

**APRIL 2022**

**ANNA UNIVERSITY :: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “BLOCK CHAIN BASED DECENTRALIZED E-COMMERCE USING ETHEREUM AND SMART CONTRACTS” is the bonafide work of “S.SANTHOSH KANNA (142219205079), R.SARANKUMAR (142219205083), T.SURYA (142219205098) and V.VIGNESHWARAN (142219205106)”who carried out the project work under my supervision.

**SIGNATURE**

**Dr .A.R. REVATHI**  
**HEAD OF DEPARTMENT**

Associate Professor  
Department of Information  
Technology  
SRM Valliammai Engineering  
College(an autonomous institution),  
SRM Nagar Kattankulathur  
Chengalpattu-603 203

**SIGNATURE**

**Mr .ASAN NAINAR**  
**SUPERVISOR**

Assistant Professor(Sel.G)  
Department of Information  
Technology  
SRM Valliammai Engineering  
College(an autonomous institution),  
SRM Nagar Kattankulathur  
Chengalpattu-603 203

**Submitted for the viva voice held on .....**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We express our sincere gratitude to our respected **Director Dr.B.Chidhambara Rajan, M.E., Ph.D.**, for his constant encouragement, which has been our motivation to strive towards excellence. We thank our sincere **Principal Dr.M.Murugan, M.E., Ph.D.**, for his constant encouragement, which has been our motivation to strive towards excellence.

We extend our hand of thanks to our Head of the Department, **Dr.A.R.Revathi,B.E.,M.Tech,M.B.A, Ph.D.**, Associate Professor for her unstinted support. We are grateful to our internal guide **Mr.Asan Nainar, Assistant Professor(Sel.G)** without whose invaluable guidance and encouragement, this project would not have been a success.

We would thank to **Dr.A.R.Revathi,B.E.,M.Tech,M.B.A, Ph.D.**, Assistant Professor (O.G) and **Mrs.SathyaDevi, B.E., M.E., Assistant Professor (O.G)** for their consistent and encouragement throughout the progress of the project.

We also like to thank all Teaching and non-teaching staff members of our department, for their support during the course of the project.

We finally thank our friends and family for their support during the course of the project

## ABSTRACT

In this project, we present a prototype of multi-user system for access control to datasets stored cloud environment. Cloud storage like any other untrusted environment needs the ability to secure share approach provides an access control over the data stored in the cloud without the provider participation. The ss control mechanism is ciphertext-policy attribute- based encryption scheme with dynamic attributes. Using a decentralized ledger, our system provides immutable log of all meaningful security events, such as key policy assignment, change or revocation, access request. We propose a set of cryptographic protocols ensuring graphic operations requiring secret or private keys. Only ciphertexts of hash codes are transferred through the . The prototype of our system is implemented using smart contracts and tested on Ethereum blockchain platform.

Database play a crucial role for any individual as well as any organization and business to store its data. Realizing the importance of data and insufficiency of storage, databases are replicated, distributed and backed up in different ways. Individuals store data in the cloud provided by different privately companies. Organizations set up their data centers at different part of the globe to store its data. For the security and bandwidth, data are scatteredand replicated to different servers at different places. This seems to provide a good solution for the management of rapidly increasing data. And also ensures data safety. In future, the rate of increment of data is sure to reach high. To cope with it, the current database system needs to be more reliable, safe and available all the time.

**Keywords-** Blockchain,Database and Ciphertext

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>iv</b>
	<b>LIST OF FIGURES</b>	<b>xxiii</b>
<b>1.</b>	<b>INTRODUCTION</b>	
	1.1 Introduction	<b>1</b>
	1.2 Objectives	<b>2</b>
	1.3 Challenges	<b>2</b>
<b>2.</b>	<b>LITERATURE SURVEY</b>	
	2.1 Literature survey summary	<b>3</b>
	2.2 Existing system	<b>4</b>
<b>3.</b>	<b>PROPOSED WORK</b>	
	3.1 System architecture diagram	<b>6</b>
	3.2 Hardware components	<b>13</b>
	3.3 Software components	<b>14</b>
	3.4 Methodology	<b>15</b>
	3.5 Modules	<b>18</b>

<b>4.</b>	<b>EXPERIMENTAL RESULTS</b>	<b>21</b>
<b>5.</b>	<b>CONCLUSION</b>	<b>24</b>
<b>6.</b>	<b>APPENDICES</b>	<b>26</b>
<b>7.</b>	<b>REFERENCES</b>	<b>42</b>

## **LIST OF FIGURES**

<b>FIG No.</b>	<b>NAME</b>	<b>PAGE No.</b>
<b>3.1</b>	<b>SYSTEM ARCHITECHTURE</b>	
	Fig No 3.1 System Architecture	6
	Fig No 3.2 Process	11
<b>3.4</b>	<b>METHODOLOGY</b>	
	Fig No 3.4 Methodology	15
<b>3.5</b>	<b>MODULES</b>	
	Fig No 3.5 File Processing	20
<b>4.0</b>	<b>EXPERIMENTAL RESULTS</b>	
	Fig No 4.1 Home Page	21
	Fig No 4.2 Login Authentication Page	21
	Fig No 4.3 Products Page	22
	Fig No 4.4 Product Description Page - 1	22
	Fig No 4.5 Product Description Page - 2	23
	Fig No 4.6 Transaction Page	23

## **1.1 INTRODUCTION**

With the growing fields of Information Technology, Internet Of Things and Digitization of every business, organizational work and projects, Information has become the biggest valuable asset for anyone. Data has become the most powerful thing in today's world. With the abundance of data and its ever growing nature, it's equally important to store it in an organized way such that it's easily accessible and secure. For this purpose Databases are being used as a warehouse to store data.

Database play a crucial role for any individual as well as any organization and business to store its data. Realizing the importance of data and insufficiency of storage, databases are replicated, distributed and backed up in different ways. Individuals store data in the cloud provided by different privately companies. Organizations set up their data centers at different part of the globe to store its data. For the security and bandwidth, data are scattered and replicated to different servers at different places. This seems to provide a good solution for the management of rapidly increasing data. And also ensures data safety. In future, the rate of increment of data is sure to reach high. To cope with it, the current database system needs to be more reliable, safe and available all the time.



## 1.2 OBJECTIVES

To develop a system over Ethereum Blockchain which can store the users data in a decentralized database distributed across the peer to peer network. The specific objectives are as follows:

- To fulfill the requirements of the partial fulfillment of the Bachelor's degree in Computer Engineering, Institute of Engineering, Tribhuvan University
- To research about cryptography, P2P network, web technology and blockchain.
- To contribute in the active research on decentralized applications and cryptography.
- To develop a distributed cloud storage platform.

## 1.3 CHALLENGES

### **Lack of Privacy of Data**

Different cloud service companies and distributed data centre of organization ensures the data availability and safety. However, most of them have terms that allow the company to edit, modify, access, delete, view and analyze your content.

### **Data Loss**

Storing sensitive data only on local machine or drives can sometime be very lamenting because once they are stolen, lost or destroyed by any other means, user cannot make a recovery. Moreover, most of the personal accounts of Cloud Storage also do not cover the insurance of data

## **2. LITERATURE REVIEW**

**Shangping Wang “A Block chain-Based Distributed Storage Network to Manage Growing Data Storage Needs” 2019IN IEEE ACCESS, VOL. 9, PP. 57426- 57439, 2021, DOI: 10.1109/ACCESS.2019.52108..**

In recent years, Counterfeit goods play a vital role in product manufacturing industries. This Phenomenon affects the sales and profit of the companies. To ensure the identification of real products throughout the supply chain, a functional block chain technology used for preventing product counterfeiting. By using a block chain technology, consumers do not need to rely on the trusted third parties to know the source of the purchased product safely. Any application that uses block chain technology as a basic framework ensures that the data content is “tamper-resistant”. In view of the fact that a block chain is the decentralized, distributed and digital ledger that stores transactional records known as blocks of the public in several databases known as chain across many networks. Therefore, any involved block cannot be changed in advance, without changing all subsequent block. In this paper, counterfeit products are detected using barcode reader, where a barcode of the product linked to a Block Chain Based Management (BCBM) system. So the proposed system may be used to store product details and unique code of that product as blocks in database. It collects the unique code from the customer and compares the code against entries in block chain database. If the code matches, it will give notification to the customer, otherwise it gets information from the customer about where they bought the product to detect counterfeit product manufacturer.

**Vijay A.Kanade “A Secure Cloud Storage Framework With Access Control Based on Block chain” 2021 pp. 1-5, DOI: 10.1109/ICETAS.2017.8277548.**

IoT industry has come to the fore in this modern techno-frenzy age. This has caused digital data explosion and has kept the data storage industry on its toes. The paper proposes a novel blockchain-based data storage model where users contribute the storage space of their personal electronic devices to meet the growing data storage needs. The paper dives into a decentralized system design, along with an equitable compensation model for all the storage space contributing users. The decentralized framework provides a common platform for interaction between the storage space contributors and buyers. The proposed solution provides an alternative to cloud storage which relies heavily on servers.

**Jiangang Shu and Xing Zou “Block chain-Based Decentralized Public Auditing for Cloud Storage” 2021 International Conference on Information and Communications Technology (ICOIACT), 2019, pp. 206-211, doi:10.1109/ICOIACT46704.2019.8938570.**

Public auditing schemes for cloud storage systems have been extensively explored with the increasing importance of data integrity. A third-party auditor (TPA) is introduced in public auditing schemes to verify the integrity of outsourced data on behalf of users. To resist malicious TPAs, many blockchain-based public verification schemes have been proposed. However, existing auditing schemes rely on a centralized TPA

### **3. PROPOSED WORK**

- In this project, we are using web 3.0 technology which is a trending domain and it is also called as block chain technology.
- We are combining cloud technology with block chain technology to enhance the security and privacy related features.
- This system provides a secure storage with the help of Ethereum network and smart contracts which plays a major role in protecting the network
- This is totally based on decentralized server which makes the user to be not dependent on a particular service or a system.
- The main objective of this project is to build a secure storage combining cloud technology and block chain
- The secure storage system can be achieved using IPFS (Inter Planetary File System)
- Block chain is considered by many to be the safest technology. Here, a single file is splited and stored in blocks using the file manipulation and Advanced Encryption Standard (AES) algorithms.
- A high level, contract oriented language, Solidity is used for writing smart contracts.
- This project combines Cloud with block chain technology to produce a secured file storage system

### 3.1 SYSTEM ARCHITECTURE

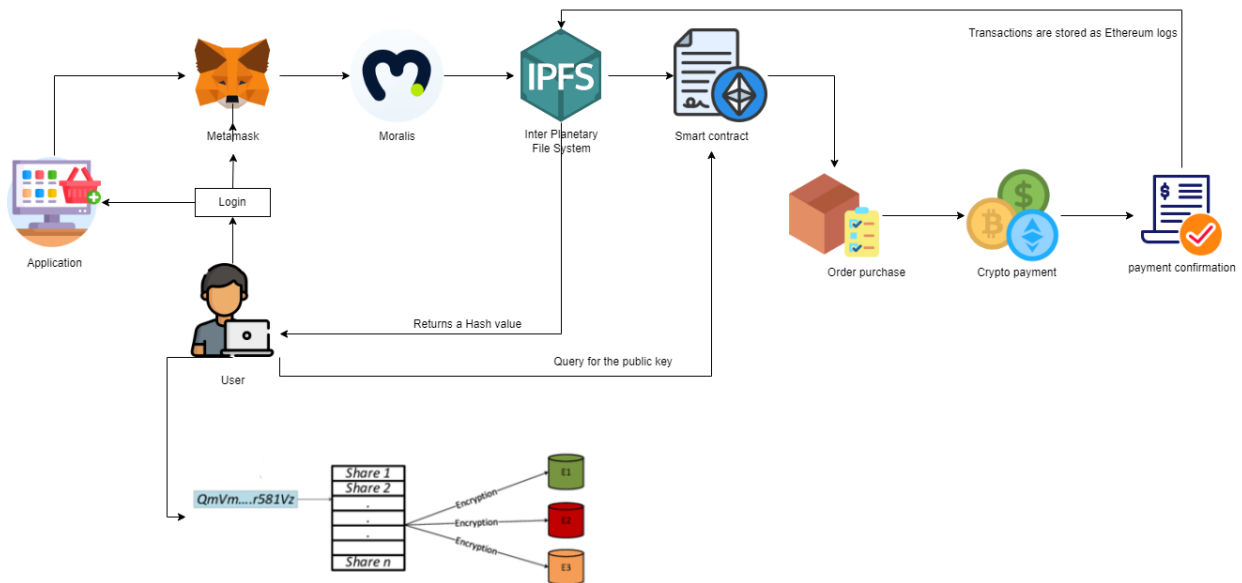


Fig No 3.1 System Architecture

The traditional architecture of the World Wide Web uses a client-server network. In this case, the server keeps all the required information in one place so that it is easy to update, due to the server being a centralized database controlled by a number of administrators with permissions.

In the case of the distributed network of blockchain architecture, each participant within the network maintains, approves, and updates new entries. The system is controlled not only by separate individuals, but by everyone within the blockchain network.

### HASHING

Hash is a unique set of characters or an array of bytes derived from a function which in takes certain message or plain text. Each message has

unique hash that depends on the hash function used to derive the hash. A slight change in any character of input will produce an entire different hash.

The hash value is always of same size in most of the cases. For our project, we will be using SHA-256 hashing. The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256bit (32-byte) hash. Hash is a one way function. It cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures. SHA256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available.

## **DISTRIBUTED HASH TABLE**

Hash Table is data structure capable of storing (key,value) pairs and performing value lookup when key is provided. The insertion and lookup of (key,value) pair is very fast in the order of  $O(1)$  as they internally use arrays to keep the data. But the array is always larger than the total no. (key,value) pair which results in slower iteration over the items in hash table.

Distributed Hash Table is a distributed service that provides (key,value) pair like Hash tables. Each node participating in the distributed network maintains only a fixed number of (Key,Value) pair. A method or rule is made such that when a key is known, the peer which might have the value can be determined. The value is asked with one or more such peers. If the peer doesn't have the value it asks with other peers until the value is retrieved.

## KADEMLIA

Kademlia is a communications protocol for peer-to-peer networks. It is one of many versions of a DHT, a Distributed Hash Table. Kademlia uses a “distance” calculation between two nodes. This distance is computed as the exclusive or (XOR) of the two node IDs, taking the result as an integer number. Keys and Node IDs have the same format and length, so distance can be calculated among them in exactly the same way. The node ID is typically a large random number that is chosen with the goal of being unique for a particular 10. It can and does happen that geographically widely separated nodes from Germany and Australia, for instance, can be “neighbors” if they have chosen similar random node IDs.

- STORE: Stores a (key, value) pair in one node.
- FIND\_NODE: The recipient of the request will return the  $k$  nodes in his own buckets that are the closest ones to the requested key.
- FIND\_VALUE: Same as FIND\_NODE, but if the recipient of the request has the requested key in its store, it will return the corresponding value.

A Kademlia network is characterized by three constants, which we call  $\alpha$ ,  $B$ , and  $k$ . The first and last are standard terms. The second is introduced because some Kademlia implementations use a different key length.

- $\alpha$  is a small number representing the degree of parallelism in network calls.
- $B$  is the size in bits of the keys used to identify nodes and store and retrieve data.
- $k$  is the maximum number of contacts stored in a bucket.

- **Node A** Kademlia network consists of a number of cooperating nodes that communicate with one another and store information for one another. Each node has a nodeID, a quasi-unique binary number that identifies it in the network. Within the network, a block of data, a value, can also be associated with a binary number of the same fixed length B, the value's key. A node needing a value searches for it at the nodes it considers closest to the key. A node needing to save a value stores it at the nodes it considers closest to the key associated with the value.
- **NodeID** NodeIDs are binary numbers of length  $B = 160$  bits. In basic Kademlia, each node chooses its own ID by some unspecified quasirandom procedure. It is important that nodeIDs be uniformly distributed; the network design relies upon this. While the protocol does not mandate this, there are possible advantages to the node's using the same nodeID whenever it joins the network, rather than generating a new, session-specific nodeID.
- **Key** Data being stored in or retrieved from a Kademlia network must also have a key of length B. These keys should also be uniformly distributed. There are several ways to guarantee this; the most common is to take a hash, such as the RIPEMD160 digest, of the value.
- **Kademlia Metrics** Kademlia's operations are based upon the use of exclusive OR, XOR, as a metric. The distance between any two keys or nodeIDs  $x$  and  $y$  is defined as  $\text{distance}(x, y) = x \hat{y}$  where  $\hat{\phantom{x}}$  represents the XOR operator. The result is obtained by taking the bitwise exclusive OR of each byte of the operands.
- **Kademlia Protocol** Kademlia has four messages.

The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data



file. SHA-256 algorithm generates an almost- unique, fixed size 256-bit (32-byte) hash. Hash is a one way function. It cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures. SHA-256 is one of the successor hash functions to SHA- 1, and is one of the strongest hash functions available.

## **BLOCKCHAIN**

Blockchain is the growing list of records called blocks, containing structured information linked using the art of cryptography distributed globally . Each block in blockchain contains the cryptographic hash of previous block along with timestamp and data which typically varies with use-cases. Merkle trees are the fundamental part of blockchain. Every block contains the block header which is outcome of recursive cryptographic hashes of all the data nodes or transaction from bottom to up approach. During hash generation, the order of data matters for the final hash of the block. If single detail of transactions or order of transaction changes then changes the merkle hash. Therefore, Merkle Root summarizes all of the data in the related 12 transactions, and is stored in the block header. This feature makes blockchain resistance to modification which is considered secure by design.

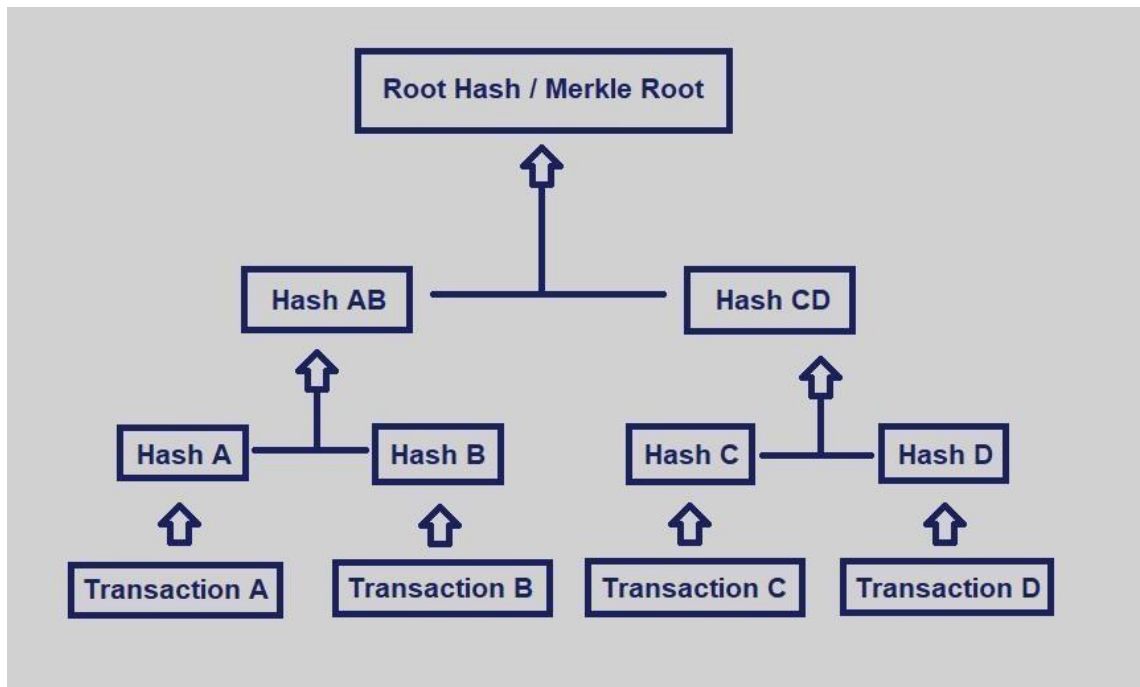


Fig No 3.2 Process

Blockchain is P2P network which relies on protocol for inter-node communication and validating the blocks. Blockchain was invented by Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the cryptocurrency bitcoin. Blocks in Blockchain are the holder of valid transactions that are hashed and encoded in Merkle tree. Every block contains the cryptographic hash of previous blocks along with its own data which therefore forms the chain. This iterative mechanism in each block conforms the integrity of previous block all the way back to genesis block.

Block time is the average time it takes in the network to generate 1 extra block in blockchain. By the time block is generated, the data of that block is verified. This means lesser the block time, faster the transactions. A hard fork is a rule change such that the software validating according to the old rules will see the blocks produced according to the new rules as invalid.

There will be no central point vulnerability, no center point of failure in blockchain. It is open to public which makes it more user-friendly than traditionally owned records. Being permissionless and open, there is no need to guard against bad actors. platform and operating system featuring smart contract (scripting) functionality. Ether is a cryptocurrency whose blockchain is generated by the Ethereum platform. Ethereum provides a decentralized Turing-complete virtual machine

Ethereum Virtual Machine (EVM), which can execute scripts using an international network of public nodes. "Gas", an internal transaction pricing mechanism, is used to mitigate spam and allocate resources on the network. Ethereum addresses are composed of the prefix "0X" a common identifier for hexadecimal, concatenated with the rightmost 20 bytes of the Keccak256 (SHA-3) hash (big endian) of the ECDSA (Elliptic Curve Digital Signature Algorithm) public key.

- Smart Contract Ethereum's smart contracts are based on different computer languages, which developers use to program their own functionalities. Smart contracts are high-level programming abstractions that are compiled down to EVM bytecode and deployed to the Ethereum blockchain for execution. They can be written in Solidity (a language library with similarities to C and JavaScript), Serpent (similar to Python, but deprecated), LLL (a low-level Lisp-like language), and Mutan (Go-based, but deprecated). There is also a research-oriented language under development called Viper (a strongly-typed Python-derived decidable language). 14

- ERC20 Token ERC-20 is a technical standard used for smart contracts on the Ethereum blockchain for implementing tokens. ERC stands for

Ethereum Request for Comment

## **3.2 HARDWARE REQUIREMENTS**

### **PERSONAL COMPUTER / LAPTOP**

A Pc/Laptop with i5 9th Generation processor or higher and then 8gb RAM or higher. These are the basic requirements for this system.

## **3.3 SOFTWARE REQUIREMENTS**

### **3.3.1 HTML:**

HTML (HyperText Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables

### **3.3.2 CSS:**

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML)

### **3.3.3 JAVASCRIPT:**

Javascript is used by programmers across the world to create dynamic and interactive web content like applications and browsers. JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by 97.0% of all websites.

### **3.3.4 TRUFFLE GANACHE:**

Truffle is a “world-class development environment, testing framework, and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to 16 make life as a developer easier.

### **3.3.5 MATIC NETWORK:**

An Indian-founded blockchain platform is providing a solution to these challenges via its innovative and unique layer 2 solution. Let us find out what it does and why it has a strong future potential. Polygon (with ticker MATIC) is a complete multi- chained system, a framework as well as a protocol.

### **3.3.6 IPFS:**

The Inter Planetary File System (IPFS) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices. PFS allows users to host and receive content in a manner similar to BitTorrent. As opposed to a centrally located server, IPFS is built around a decentralized system[5] of user-operators who hold a portion of the overall data, creating a resilient system of file storage and sharing. Any user in the network can serve a file by its content address, and other peers in the network can find and request that content from any node who has it using a distributed hash table (DHT).

### 3.4 METHODOLOGY

#### Building P2P network

#### Peer to Peer Network

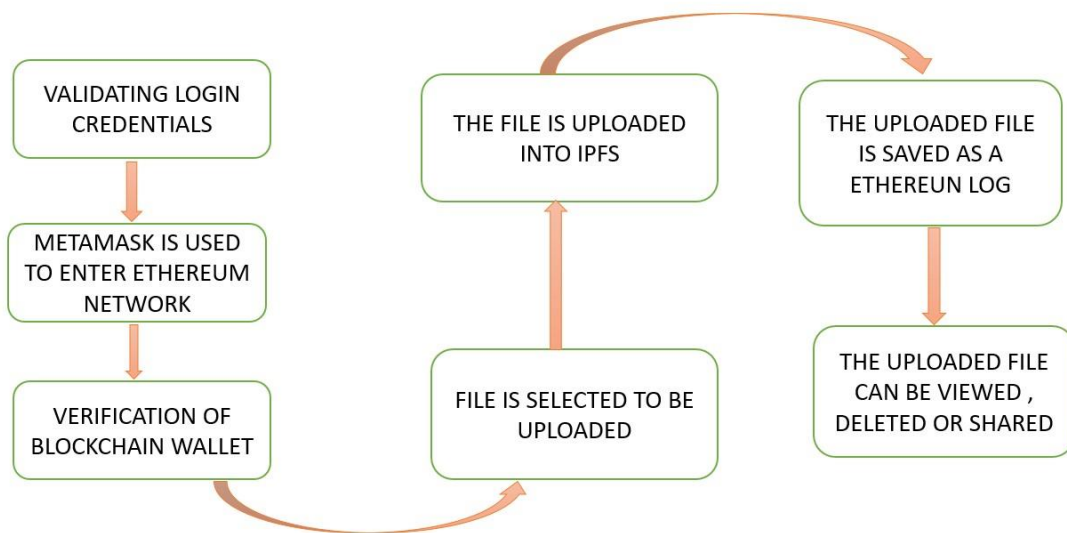


Fig No 3.4 Methodology

Peer to Peer network is the distributed network where each node in the network communicates with each other directly or through a series of channels via other nodes. There is no client server to access the resource. Each node will act both as a host or a client as needed. There is no any Central server for controlling the system flow and other nodes.

#### Message Dispatcher

Message Dispatcher class is responsible for sending and receiving messages from network. Message for each different purpose is available as a extension of MessageClass. The message is serialized in a predefined format and a TCP/Udp packet is made using the serialized data. Then it is sent to the respective destination. MessageDispatcher handles everything required for concurrently sending and receiving messages in a separate thread Pool. We have written both UDP based

and TCP based Message Dispatcher classes, but we later chose to use the UDP based Dispatcher as there was no need to setup connections. Each message was represented by a byte array thus we didn't need the stream based protocol that TCP provided.

Message Dispatcher class also performs the task of updating Contact Bucket when a node sends message. It reports when a new connection is received and when the ip address of a node changes. TimeStamped Store TimeStamped Store is the HashTable part of DHT. TimeStamped Store keeps the (key,value) pairs along with the entry and expiry time. The "key" part is obtained from RIPEMD160 of the original key. Like the Contact Bucket, TimeStamped Store contains 160 different slots labeled by unique index 0 - 159 . The slot in which (key,value) pair goes is determined by the xor based arithmetic between id of the node and they value of key.

## **Operation In DHT**

– Find Node As name suggests, find node fetches information of a node based on the node ID. If the information is available in local Contact Bucket, the information is simply returned. Otherwise, the information must be fetched from other nodes in the DHT. The purpose of xor distance based positioning in Contact Bucket serves for this purpose. When searching for a node in DHT, we follow following steps.

- \* select alpha A no of nodes from our Contact Bucket that are closest to the required node. 19
- \* Ask each of nodes to return alpha nodes closest to the required node in their bucket. The message sent is serialize as 'FindNode' message

\* From the newly returned list of nodes, we again select alpha no. of nodes that are closest to the required node and yet not queried. The returned nodeList is serialized as 'NodeListMessage'. Above process is repeated until we find the node, or we are out of nodes to query. In each iteration we find more and more closer The project consists of a P2P network where a node can join the network and provide the storage services for the client.

The system uses several Cryptography and Network algorithms and provides two major services to client :

Secure File Storage and Secret Sharing. The agreement between the parties are bound by Smart Contract and uses ERC20 token as a value for service. The two layers of the system can also be implemented separately as components for different use cases. The InterPlanetary File System (IPFS) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content addressing to uniquely identify each file in a global namespace connecting all computing devices.

IPFS allows users to host and receive content in a manner similar to BitTorrent. As opposed to a centrally located server, IPFS is built around a decentralized system of user-operators who hold a portion of the overall data, creating a resilient system of file storage and sharing. Any user in the network can serve a file by its content address, and other peers in the network can find and request that content from any node who has it using a distributed hash table (DHT).



## **3.5 MODULES**

In this project, we are using web 3.0 technology which is a trending domain and it is also called as block chain technology. We are combining cloud technology with block chain technology to enhance the security and privacy related features.

- MODULE 1: Products Page
- MODULE 2: Cryptocurrency Purchase
- MODULE 3: Customer Data Management
- MODULE 4: Admin & Backend

### **3.5.1. Products Page Module**

The submitted data will be validated in this module. Following validation, the student will get an email with a login credential from the administrator, allowing them to check the status of their allocation progress. In order to maintain a solid student database, admin able to verify all the details of the registered student, update the student's detail, delete a student from database. The main functionality of the admin portal is to manage the room allotment process, maintaining the database, providing login credentials and registration process, viewing suggestions, queries, vacate intimation from the student. This module handles the room assignment procedure automatically by a backend algorithm.

### **3.5.2 Cryptocurrency Purchase Module**

Home pages are located in the root directory of the website. Many home pages act as a virtual directory for a site — they provide top-level menus where visitors can go deeper into various areas of the site. For instance, a

typical website has a homepage with menu items like “about,” “contact,” “products,” “services,” “press” or “news.”

In addition, the home page often serves to orient visitors by providing titles, headlines and images and visuals that show what the website is about, and in some cases, who owns it and maintains it. One of the best examples is the average business website, which has the business name in a prominent place, and often features the logo, while also showing pictures related to that business, for instance, who works there, what the business produces, or what it does in a community.

### **3.5.3. Customer Data Management Module**

Blockchain storage is a way of saving data in a decentralized network, which utilizes the unused hard disk space of users across the world to store files. The decentralized infrastructure is an alternative to centralized cloud storage and can solve many problems found in a centralized system.

Blockchain relies on distributed ledger technology (DLT). The DLT acts as a decentralized database of information about transactions between various parties. Operations fill the DLT in chronological order and are stored in the ledger as a series of blocks. An interconnected chain is formed between blocks with each one referring to the block before it, thus creating a Blockchain.

### **3.5.4 Admin & Backend Module**

Storing a whole document on-chain is possible with certain blockchains, however, it is rarely a good idea. Due to the huge data demands, unless it is a very small file or of extreme importance, you would be better choosing another method. If you wanted to store the document on Bitcoin, then you first have to compress it and then format it into a hexadecimal form.

The problem with storing whole documents on a blockchain is because of something called access latency. This just means how long it takes network users to upload and download files, such as documents. Fully decentralized public blockchains have thousands of nodes. Unfortunately, the benefits that come with this number of nodes also results in a corresponding increase in latency. Any file storage, including documents, needs to have low latency otherwise the system becomes clogged up, slow, and expensive to use.

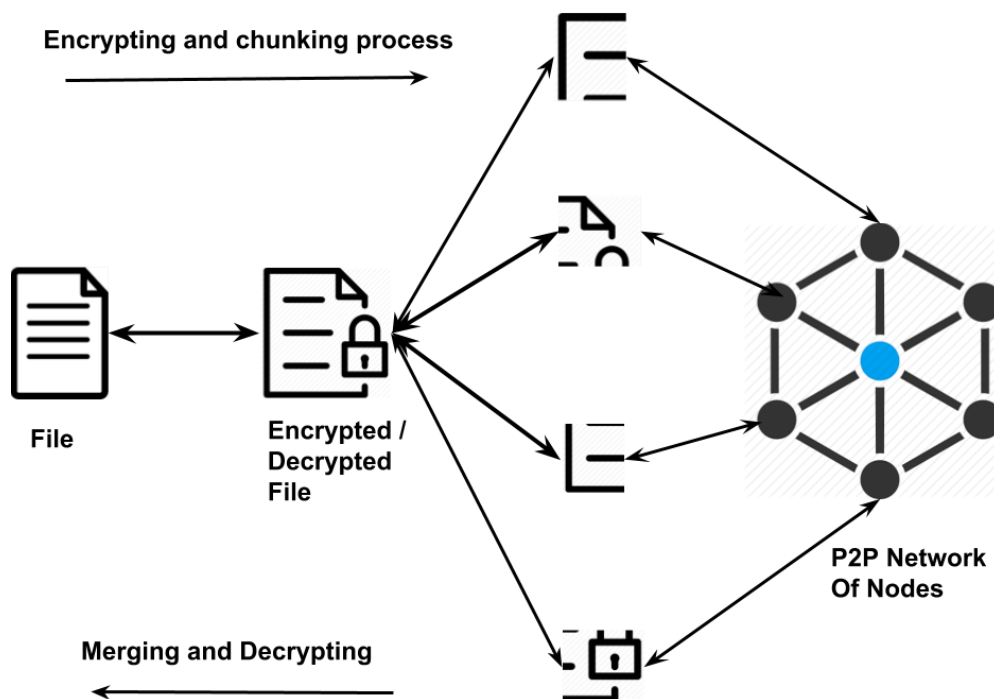


Fig No. 3.5 File Processing

## 4. EXPERIMENTAL RESULTS

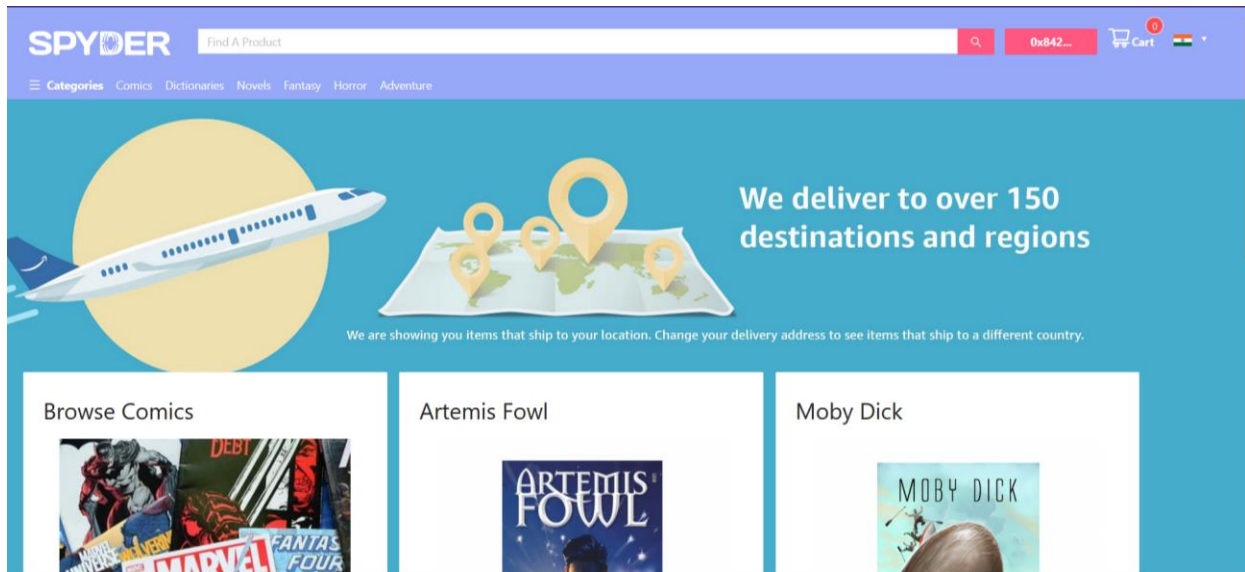


Fig No. 4.1 Home Page

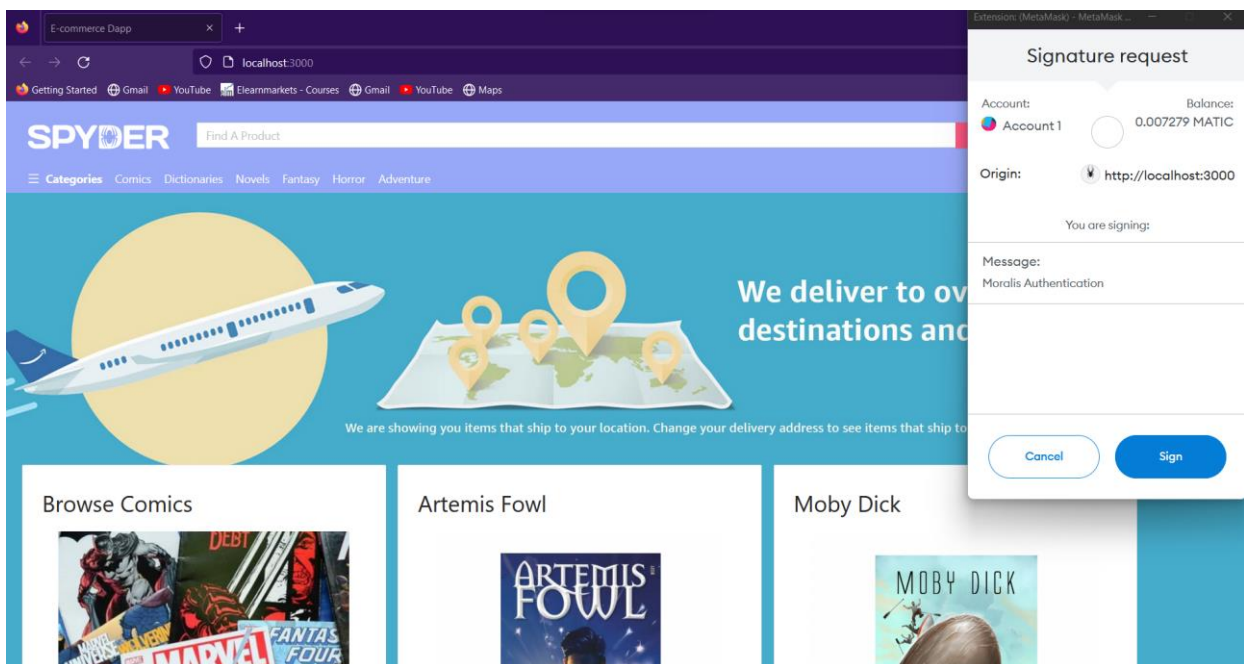
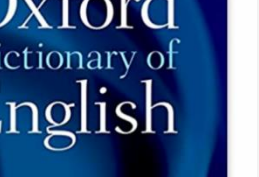


Fig No. 4.2 Login Authentication





Hover over image to zoom

## Oxford English Dictionary (3rd Edition)

★★★★★

Price: **\$65.29**

No Import Fees & Free Shipping Included

---

### About This Item

The foremost single volume authority on the English language, the Oxford Dictionary of English is at the forefront of language research, focusing on English as it is used today. It is informed by the most up-to-date evidence from the largest language research programme in the world, including the two-billion-word Oxford English Corpus.

**\$65.29**

No Import Fees & Free Shipping Included

**In Stock**

Quantity

1

22

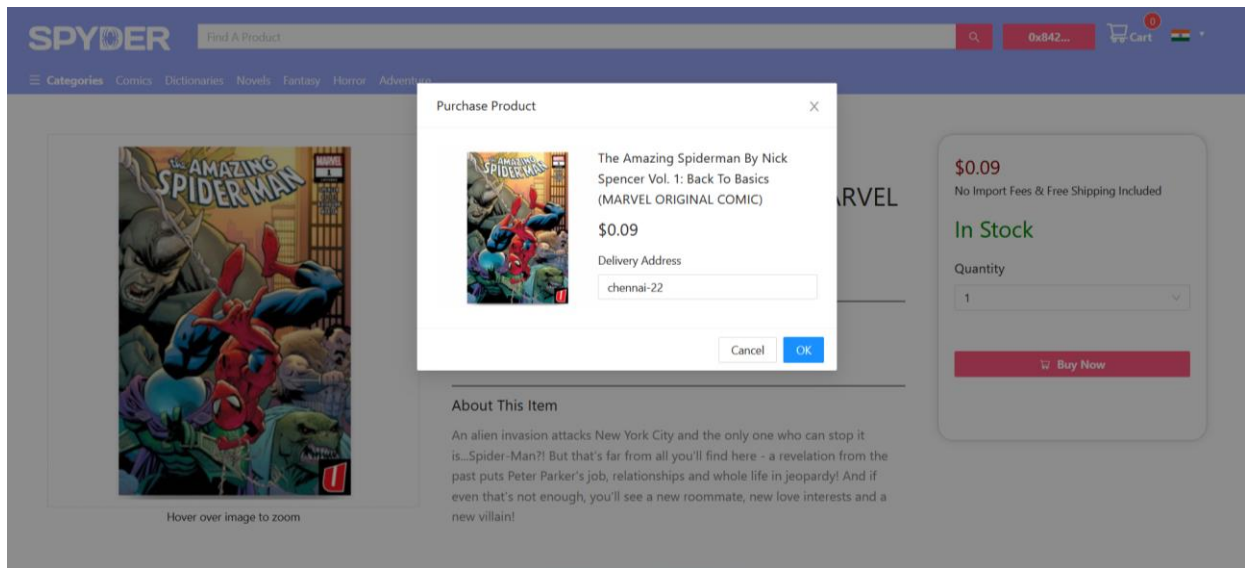


Fig No. 4.5 Product Delivery Page

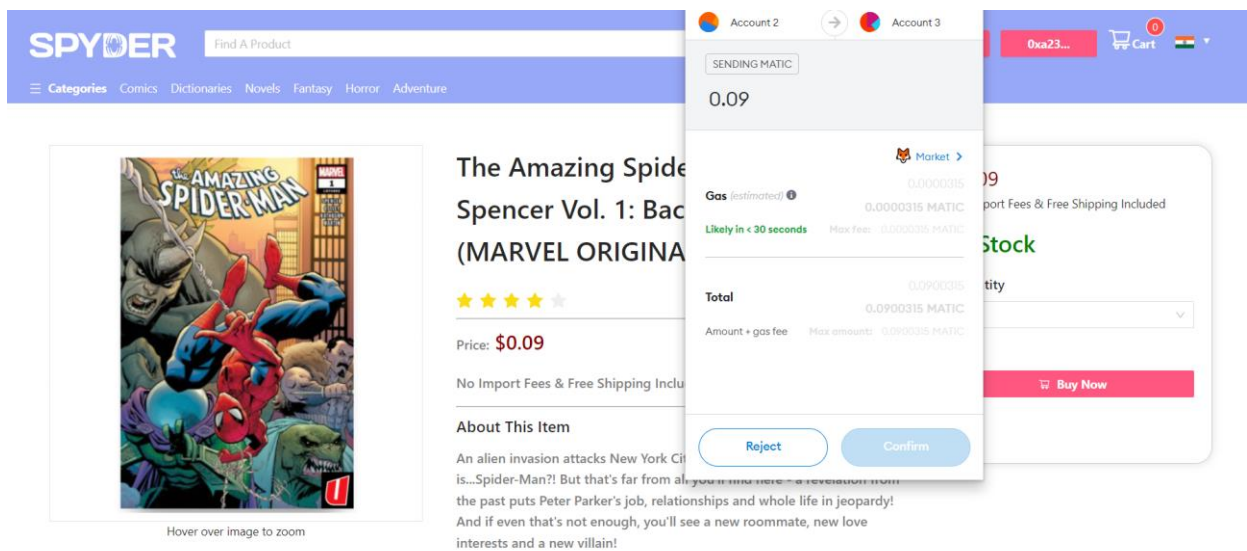


Fig No. 4.6 Transaction Page

## 5. CONCLUSION

The project consists of a P2P network where a node can join the network and provide the storage services for the client. The system uses several Cryptography and Network algorithms and provides two major services to client: Secure File Storage and Secret Sharing. The agreement between the parties are bound by Smart Contract and uses ERC20 token as a value for service. The two layers of the system can also be implemented separately as components for different use cases. The project was completed with a exciting exploration and research in Cryptography and Blockchain field. There is always room for the improvements in any projects. The project can be further enhanced including following features:

- Currently, we have only mobile app for clients to use. Therefore, web app can be made for client purposes so that large sized files can be easily encrypted, splitted and merged.
- Compensation mechanism for faulty party can be further added in our system. The one approach for this could be the introduction of a central authority. Or it can also be achieved by using a third auditor node selected at random from the network.
- Algorithms and the protocols used in the cryptography processes could further be fine-tuned. With the growing fields of Information Technology, Internet of Things and Digitization of every business, organizational work and projects, Information has become the biggest valuable asset for anyone.

Data has become the most powerful thing in today's world. With the abundance of data and it's ever growing nature, it's equally important to store it in an organized way such that it's easily accessible and secure.

For this purpose Databases are being used as a warehouse to store data. Database play a crucial role for any individual as well as any organization and business to store its data. Realizing the importance of data and insufficiency of storage, databases are replicated, distributed and backed up in different ways. 24 Individuals store data in the cloud provided by different privately companies. Organizations set up their data centers at different part of the globe to store its data.

For the security and bandwidth, data are scattered and replicated to different servers at different places. This seems to provide a good solution for the management of rapidly increasing data. And also ensures data safety. In future, the rate of increment of data is sure to reach high. To cope with it, the current database system needs to be more reliable, safe and available all the time.

The project consists of a P2P network where a node can join the network and provide the storage services for the client. The system uses several Cryptography and Network algorithms and provides two major services to client.

Secure File Storage and Secret Sharing. The agreement between the parties are bound by Smart Contract and uses ERC20 token as a value for service. The two layers of the system can also be implemented separately as components for different use cases.



## 6.APPENDICIES

### SOURCE CODE:

#### HTML:

```
import React, { Component } from 'react';

import { convertBytes } from './helpers';

import moment from 'moment' class Main
extends Component { render() { return (

<div className="container-fluid mt-5 text-center">

<div className="row">

<main role="main" className="col-lg-12 ml-auto mr-auto" style={{
maxWidth: '1024px' }}>

<div className="content">

<p>&nbsp;</p>

<div className="card mb-3 mx-auto bg-dark" style={{ maxWidth:
'512px' }}>

<h2 className="text-white text-monospace
bgdark"><b><ins>Share File</ins></b></h2>

<form onSubmit={(event) => {
event.preventDefault() const description =
this.fileDescription.value
this.props.uploadFile(description)
}}
```

```

<div className="form-group">

  <br></br> <input id="fileDescription"

  type="text" ref={(input) => {

    this.fileDescription = input }}

  className="form-control text-monospace"

  placeholder="description..." required />

</div>

<input type="file" onChange={this.props.captureFile} className="text-
white text-monospace"/>

<button          type="submit"          className="btn-primary
btnblock"><b>Upload!</b></button>

</form>

</div>

<p>&nbsp;</p>

<table className="table-sm table-bordered text-monospace" style={{
width: '1000px', maxHeight: '450px'}}>

  <thead style={{ 'fontSize': '15px' }}>

    <tr className="bg-dark text-white">

      <th scope="col" style={{ width: '10px'}}>id</th>

      <th scope="col" style={{ width: '200px'}}>name</th>

      <th scope="col" style={{ width: '230px'}}>description</th>

      <th scope="col" style={{ width: '120px'}}>type</th>

```

```

<th scope="col" style={{ width: '90px'}}>size</th>

<th scope="col" style={{ width: '90px'}}>date</th>

<th scope="col" style={{ width: '120px'}}>uploader/view</th>

<th scope="col" style={{ width: '120px'}}>hash/view/get</th>

<th scope="col" style={{ width: '120px'}}>delete</th>

</tr>

</thead>

{ this.props.files.map((file, key) => { return(

<thead style={{ 'fontSize': '12px' }} key={key}>

<tr id="row1">

<td>{file.fileId}</td>

<td>{file.fileName}</td>

<td>{file.fileDescription}</td>

<td>{file.fileType}</td>

<td>{convertBytes(file.fileSize)}</td>

<td>{moment.unix(file.uploadTime).format('h:mm:ss A

M/D/Y')}</td>

<td>

<tr className="bg-dark text-white">

<th scope="col" style={{ width: '10px'}}>id</th>

<th scope="col" style={{ width: '200px'}}>name</th>

<th scope="col" style={{ width: '230px'}}>description</th>

<th scope="col" style={{ width: '120px'}}>type</th>

```

```

<th scope="col" style={{ width: '90px'}}>size</th>

<th scope="col" style={{ width: '90px'}}>date</th>

<th scope="col" style={{ width: '120px'}}>uploader/view</th>

</tr> <a href={"https://etherscan.io/address/" +
file.uploader} rel="noopener noreferrer"
target="_blank">
{file.uploader.substring(0,10)}...

</a>

</td>

<td> <a href={"https://ipfs.infura.io/ipfs/" +
file.fileHash} rel="noopener noreferrer"
target="_blank">
{file.fileHash.substring(0,10)}...

</a>

</td>

<td> <button onclick = "Del()">

Delete

</button>

</td>

</tr>

</thead>

)

```

```
}}}
```

```
</table> </div>
```

```
</main>
```

```
</div>
```

```
</div>
```

```
);
```

```
}
```

```
} export default
```

```
Main;
```

## **CSS:**

```
.link {
```

```
font-size: 18px;
```

```
margin:5%;
```

```
}
```

```
.link:hover {
```

```
color: #A8E890;
```

```
text-decoration:  
underline;  
  
}
```

```
.carousel {  
  
position: absolute;  
  
left: 0;  
  
top: 0;  
  
}
```

```
.carousel-img {  
  
min-width: 100vw;  
  
height: 750px;  
  
object-fit: cover;  
  
}
```

```
.card {  
  
    width:450px;  
  
    height:530px;  
  
}
```

```
.card-content {  
  
    display: flex;  
  
    flex-wrap: wrap;  
  
    justify-content:  
space-between;  
  
    margin-left: auto;  
  
    margin-right: auto;  
  
    width: 90%;  
  
}
```

```
.card-category {  
  
  width: 50%;  
  
}
```

```
.cards {  
  
  display: flex;  
  
  justify-content:  
start;  
  
  flex-wrap: wrap;  
  
  position: absolute;  
  
  left: 0;  
  
  top: 450px;  
  
  padding-left: 20px;  
  
  padding-right:  
20px;  
  
  width: 100vw;
```



```
gap: 15px;

}

.results-header {

border-top: 1px
solid #ccc;

border-bottom: 1px
solid #ccc;

box-shadow: 0 0
10px #ddd;

font-size: 16px;

padding: 10px
25px;

font-weight: 500;

}
```

```
.category {  
  
  color: #EA047E;  
  
  font-weight: bold;  
  
}
```

### **JAVASCRIPT:**

```
import React from  
'react';  
  
import { Link } from  
"react-router-dom";  
  
import Header from  
"../components/Hea  
der";  
  
import  
"./Home.css";  
  
import { Carousel,  
Card } from 'antd';
```

```
import Carousel1  
from  
"../images/carousel1  
.png";
```

```
import Carousel2  
from  
"../images/carousel2  
.png";
```

```
import Carousel3  
from  
"../images/carousel3  
.png";
```

```
import Comics from  
"../images/comics.p  
ng";
```

```
import ArtemisFowl  
from  
"../images/ArtemisF  
owl.png";
```

```
import MobyDick
from
"../images/MobyDic
k.png";
```

```
import Adventure
from
"../images/adventure
.png";
```

```
import Dictionaries
from
"../images/dictionari
es.png";
```

```
import Fantasy from
"../images/fantasy.p
ng";
```

```
import Horror from
"../images/horror.pn
g";
```

```
<img  
src={ Comics }  
alt="Comics  
Category"  
className="card-  
content"></img>
```

```
<br />
```

```
<Link  
to="/categories"  
state={ "Comics" }  
className="link">
```

Shop Now

```
</Link>
```

```
</Card>
```

```
<Card  
className="card">
```

```
<h1>Artemis  
Fowl</h1>
```

```
<img  
src={ArtemisFowl}  
alt="Artemis Fowl"  
className="card-  
content"></img>
```

```
<br />
```

```
<Link to="/"   
className="link">
```

View Product

```
</Link>
```

```
</Card>
```

```
<Card  
className="card">
```

```
<h1>Moby  
Dick</h1>
```

```
<img  
src={MobyDick}  
alt="Moby Dick"
```

```
className="card-  
content"></img>
```

```
<br />
```

```
<Link    to="/"   
className="link">
```

```
    View Product
```

```
</Link>
```

```
</Card>
```

```
<Card   
className="card">
```

```
    <h1>Shop  By  
Category</h1>
```

```
    <div   
className="card-  
content">
```

```
{ catCard.map((e)
```

```
=> {
```

```
    <Link to="/"
className="link">
```

Shop All

```
</Link>
```

```
</div>
```

```
</Card>
```

```
</div>
```

```
</div>
```

```
</>
```

```
)
```

```
}
```

```
export default
```

```
Home;
```



## 7. REFERENCES

- [1] Shangping Wang “A Block chain-Based Distributed Storage Network to Manage Growing Data Storage Needs” 2019IN IEEE ACCESS, VOL. 9, PP. 57426- 57439, 2021, DOI: 10.1109/ACCESS.2019.52108..
- [2] Jiangang Shu and Xing Zou “Block chain-Based Decentralized Public Auditing for Cloud Storage” 2021 International Conference on Information and Communications Technology (ICOIACT), 2019, pp. 206-211, doi:10.1109/ICOIACT46704.2019.8938570.
- [3] Vijay A.Kanade “A Secure Cloud Storage Framework With Access Control Based on Block chain” 2021 pp. 1-5, DOI: 10.1109/ICETAS.2017.8277548.
- [4] A. Martin-Lopez, "AI-Driven Web API Testing," 2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2020, pp. 202- 205.
- [5] D. R. Ignatius Moses Setiadi, A. Faishal Najib, E. H. Rachmawanto, C. Atika Sari, K. Sarker and N. Rijati, "A Comparative Study MD5 and SHA1 Algorithms to Encrypt REST API Authentication on Mobile-based Application," 2019 International Conference on Information and Communications Technology (ICOIACT), 2019, pp. 206-211, doi:10.1109/ICOIACT46704.2019.8938570.
- [6] V. Atlidakis, P. Godefroid and M. Polishchuk, "RESTler: Stateful REST API Fuzzing," 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), 2019, pp. 748-758, doi: 10.1109/ICSE.2019.00083.

- [7] A. Neumann, N. Laranjeiro and J. Bernardino, "An Analysis of Public REST Web Service APIs," in IEEE Transactions on Services Computing, vol. 14, no. 4, pp. 957-970, 1 July-Aug. 2021, doi: 10.1109/TSC.2018.2847344
- [8] T. Nandanwar, P. Bahutule and R. Buddala, "A Study on Shift towards Digitization of Hostel Room Allotment for a University," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-4, doi: 10.1109/icETITE47903.2020.117.
- [9] I.Ahmad, E.Suwarni, R. I. Borman, Asmawati, F. Rossi and Y. Jusman, "Implementation of RESTful API Web Services Architecture in Takeaway Application Development," 2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS), 2021, pp. 132- 137, doi: 10.1109/ICE3IS54102.2021.9649679.