Decentralized Application using Blockchain

(Content Resource Management)

Samar Mahato
Dept. of Information Technology
Guru Ghasidas Vishwavidyalaya
Bilaspur, Chhattisgarh, India

Arvind Kumar
Dept. of Information Technology
Guru Ghasidas Vishwavidyalaya
Bilaspur, Chhattisgarh, India

Atul Kumar
Dept. of Information Technology
Guru Ghasidas Vishwavidyalaya
Bilaspur, Chhattisgarh, India

Prof. Deepak Kant Netam
Asst. Prof. Dept. of Information Technology
Guru Ghasidas Vishwavidyalaya
Bilaspur, Chhattisgarh, India

Abstract*— Blockchain is transforming the internet so that it now allows for the transmission of value. Blockchain can be applied in areas of the real world where we need immutability, transparency, and trust. Blockchain is a peer-to-peer network that uses consensus to create a decentralised, distributed, immutable public/private database that stores transactions as individual blocks. Everywhere, including healthcare, supply chain management, education, finance, and voting, it is applicable. In this work, we design a block chain based data storage and access framework Student Support Progression Application (CRM) to remove its total dependence on a centralized repository. We use the public block chain and tools like Moralis, Metamask, Truffle and Remix IDE for deploying the contracts. In the proposed work, metadata of the files are stored on the block chain and we use the networks like Mumbai Testnet (Polygon) for occurring the Transactions using a peer -to-peer networks. This will provide decentralized storage, distributed processing, and efficient lookup capabilities.

Keywords—Blockchain, IPFS, Decentralized System, Polygon Network, Moralis, Smart Contract.

I. INTRODUCTION

Satoshi Nakamoto first introduced blockchain in 2008. Blockchain is an online ledger that allows for decentralised and transparent data sharing. A blockchain is a distributed, peer-to-peer database that hosts an ever-increasing number of transactions. Each transaction, known as a "block," is encrypted, timestamped, and validated by every authorised member. A transaction that has not been validated by all database members is not added to the database. Every transaction is sequentially attached to the previous transaction, forming a transaction chain (or blocks). A transaction cannot be deleted or edited, resulting in an unchangeable audit trail. Only by adding another transaction to the chain can a transaction be changed.

As a result, only authenticated and authorised users can access the network, increasing system security as required by enterprise applications. Health, government services, supply chain management, Internet of Things, peer-to-peer cloud storage, and many other non-financial areas that leverage the opportunities of permissioned blockchains are just a few examples. P2P cloud storage is an intriguing blockchain application because it provides a decentralised data storage facility without the involvement of a trusted third party or a client-server architecture. Decentralized data storage will help to eliminate the most common data failures and outages by improving data security, privacy, and control.

So, why should you use Blockchain? Because the data in a Blockchain cannot be changed under normal circumstances. Even if data is changed, it only takes a second for us to detect the tampering. A data or a node in Blockchain is validated only when multiple parties approve it. As a result, the system would be Reliable and Authenticated at all times. This proposed system not only closes the gaps in our current system, but also provides us with a practical and concrete solution.

II. LITERATURE SURVEY

Blockchains are write-only data structures with no administrative access to edit or delete data. The data structures are known as blocks, and they are distributed through a peer-to-peer network. Each block contains the previous block's cryptographic hash function and is used to create a link between them. Because the linked blocks form a complete chain, the term blockchain was coined. The hash function ensures the blockchain's security, integrity, and immutability [1].

A blockchain is a distributed, shared, fault-prone, and append-only database that attempts to manage records in blocks. Blockchain does not need to be trust-based for any type of transaction. One of its features is that it solves the problem of double spending. Furthermore, as long as the valid core of the system manipulates the power of the CPU, the recording of any transaction is not transferable to a third party. Nodes can join or leave the network when necessary. When a valid block is detected by achieving a majority by voting with CPU power, the others are said to be invalid. Blockchain enables a wide range of applications, including Decentralized Applications, cross-border payments, asset management, supply chain, and logistics [2][3].

Blockchain can be used as a key exchange protocol for the document known as Blockchain-based Authenticated Key Exchange Protocol (BAKE). This

can meet the security and computational requirements of scenarios in which strangers must bootstrap trust in an untrustworthy environment. On the basis of privacy protection, multiple unfamiliar parties authenticate common secret holders via Blockchain transactions and complete the exchange of session keys [4].

Along with crypto currency and certification systems, this technology can be used in cyber security, health care, citizen identity management, supply chain product tracking, finance industry, and many other areas. This is due to its secure and transparent information management. [5]

III. PROPOSED METHODOLOGY

In this Paper, The smart contract was deployed using the Windows operating system. The Truffle tool is used to deploy the smart contract. The project we proposed is more secure than a standard Dapp. Decentralized application with increased security and integrity. Only authorised users can create accounts in this application and send messages from one account to another, as well as add. Matics are the primary tokens used to carry out all transactions on the Block Chain. Gas is the computational unit in Ethereum that is used to execute transactions. The first step in deploying smart contracts is to compile and migrate them.

1) Blockchain: Blockchain can better be understood as an immutable database and laid the foundation of the whole project. It provides a trusted environment where actions have done are visible and can't be tampered with.

2) Ethereum: Ethereum is a decentralized open-source Blockchain featuring smart contract functionality. Ethereum is itself the best example of Blockchain and its a cryptocurrency system which is the most widely used and is the next expensive cryptocurrency after bitcoin.

3) SmartContract: Smart contracts piece of code that runs on a Blockchain when a user performs some action. A Smart contract is written in many different languages including lowlevel languages like C++, Java, and high-level languages like Solidity which is closely similar to Typescript.

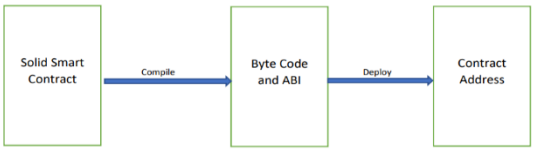

Figure – 01
Workflow of the Smart Contract

Smart contract deployment and client access the deployed contract through address and ABI.

4) Solidity: Solidity is an object-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various Blockchain platforms, most notably, Ethereum. It is closely similar to Typescript but with more specific data types.

5) IPFS: The InterPlanetary File System is a 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.

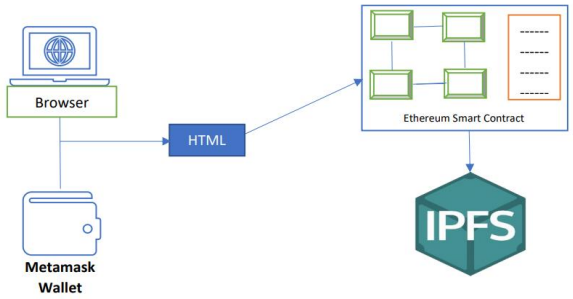

Figure - 02
Flow Diagram to connect IPFS

6) Metamask: MetaMask is an extension for accessing Ethereum enabled distributed applications or "Dapps" in your browser. The extension injects the Ethereum web3 API into every website's javascript context so that Dapps can read from the blockchain.

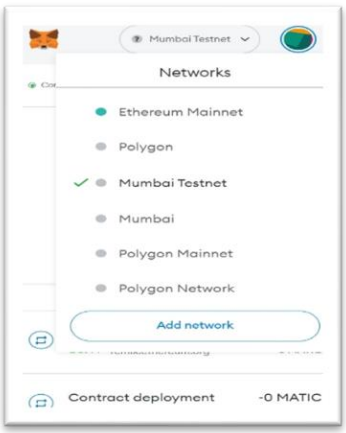

Figure – 03
Mumbai Testenet

7) Truffle: Truffle provides easy compilation, linking, deployment, and binary management of smart contracts written in solidity language.

8) Node JS: Node JS is used to write backends and is responsible for serving frontend pages, assets and managing user authentication using JWT(Json Web Token). It also has web3 as a dependency which allows us to run solidity code on frontend.

9) React: React is used to write our frontend and serves a purpose of providing a better user experience in the frontend for the end user as it provides functionality like no page reload on page switch and fast loading of sites. Keeping security in mind we have added Next.js which compiles and saves html pages in the backend and provides fast user experience along with better SEO for react pages and security as it doesn't reveal any backend details on the user end.

10) Moralis : Moralis is a service that aggregates various tools and APIs and acts as a glue between them all. All that is required when developing a decentralised app that requires blockchain access, such as Ethereum or Binance Smart Chain.

IV. PROPOSED WORK

Decentralized Dapp is an application in which users can create accounts on this platform and add tweets, delete posts, and send messages from one account to another. Because block chain is a decentralised technology that occurs all transactions with the help of matics, we use networks such as Mumbai Testnet and Polygon with matics to occur all transactions. We used Remix IDE, truffle, Polygon, and Metamask to deploy the contracts in this paper. The backbone of any blockchain project is its contract, contract is a code that runs on an ethereum node. This code is written in Solidity Language which is a high-level language which is derived from Javascript and strongly typed languages and is mainly used to write contracts.

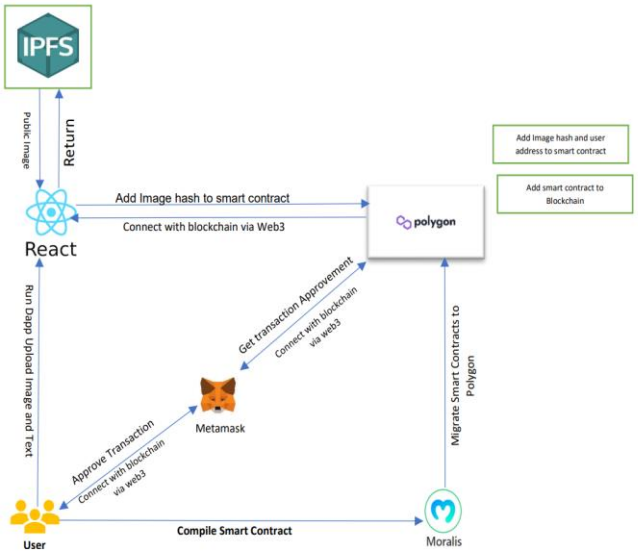

Figure - 03
Workflow Diagram of the System.

IV. IMPLEMENTATION DETAILS

A. DApp Setup Requirements

Tools and Technology used in DIUcerts DApp:

- Blockchain Framework: Ethereum
- Language for implementing smart contracts: Solidity
- IDE for deploying smart contracts: Remix IDE
- Ethereum wallet: Metamask
- Blockchain Network: Mumbai Testnet(Polygon) Network
- Front-end: React.JS
- Web Technology: Web 3.0
- Web 3.0 Module: Web3.JS library, web.th

B. Setup Metamask Account

To use metamask wallet, we have to install it via chrome extension. After the installation process, we can see the option like fig-02. If you have a previous wallet, then choose import a wallet else create a new wallet. We have added some faucet (test-net ethereum) [7][8].

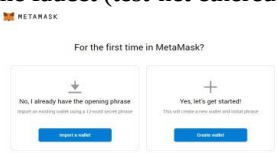

Figure- 05
Set Metamask Account

C. Deployment of Smart Contracts

To deploy the smart contract, we choose Remix IDE. Here is Fig. 03 shows the deployed smart contract of this paper in remix IDE which is web3 environment supported.

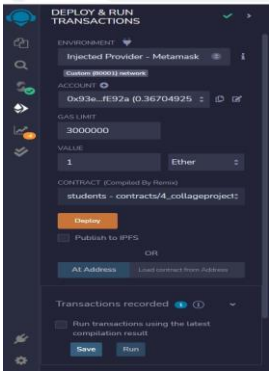

Figure – 06
Deployment of Smart Contracts in Remix IDE

D. Inject web3.js on DApp

Web3.js is a perfect and convenient way to interact with the ethereum Blockchain network. To integrate the web3.js to this paper, this can be performed using the following command in git, cmd, or terminal: [6]

• npm install web3

E. Front-end User Interface

We created a front end user interface for the application where the issuer, recipient, and verifier can easily interact with the Blockchain. We have used html, css, bootstrap, react JavaScript library to build the frontend user interface.

F. Building Smart Contract

Solidity language is used for writing and developing the smart contracts in DApp applications. Remix is extremely suitable platforms for creating, managing, testing and deploying of the smart contracts. When we used Remix IDE in the web browser as a web application, we can also use Visual Studio Code in offline on a computer. These two are the perfect combination for building Decentralized Applications [9].

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

contract students {

    address public owner;
    uint256 private counter;

    constructor() {
        counter = 0;
        owner = msg.sender;
    }

    struct data {
        address ggv_crm;
        uint256 id;
        string question;
        string Image_doc;
    }

    event dataCreated (
        address ggv_crm,
        uint256 id,
        string question,
        string Image_doc
    );

    mapping(uint256 => data) student;

    function addData(
        string memory question,
        string memory Image_doc
        ) public payable {
            require(msg.value == (0.001 ether), "Please
submit 1 Matic");
            data storage newData = student[counter];
            newData.question = question;
            newData.Image_doc = Image_doc;
            newData.ggv_crm = msg.sender;
            newData.id = counter;
            emit dataCreated(
                msg.sender,
                counter,
                question,
                Image_doc
            );
            counter++;

            payable(owner).transfer(msg.value);
    }

    function getData(uint256 id) public view returns (
            string memory,
            string memory,
            address
         ){
        require(id < counter, "No such data");

        data storage t = student[id];
        return (t.question, t.Image_doc, t.ggv_crm);
    }
}
```

IV. RESULTS

After deploying the contract need to run that particular smart contract with the help of lite-server having the local host as 3000. The contract runs and it opens in the browser and are able to perform the functions like creating an

account in twitter, adding and deleting the tweets, sending messages from one account to another account

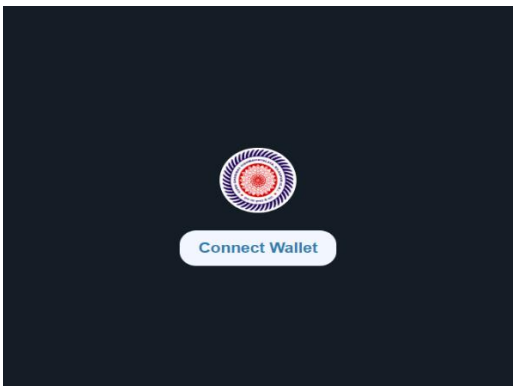

Figure-07
Connect the wallet of Metamask

A preview will be generated for the Document. If the issuer approves it the IPFS will run its hashing algorithm and store it. The same hash will be forwarded to Blockchain Node and the issuer needs to approve the Document Generation charges in Metamask and it will be saved in the Blockchain.

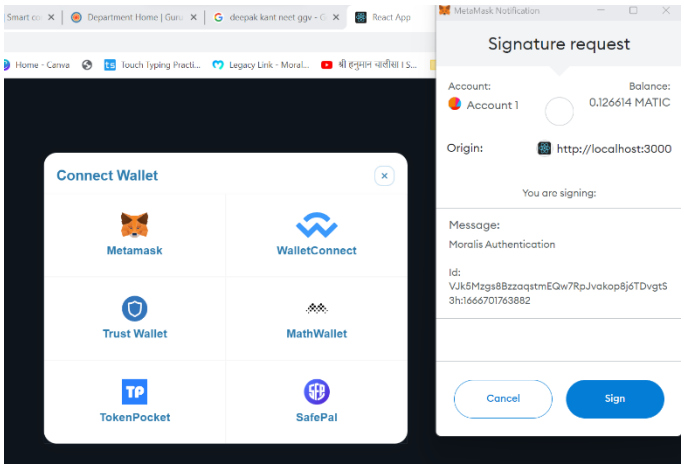

Figure-08
Metamask Notification for Sign in.

Figure – 09
Upload Image and Text

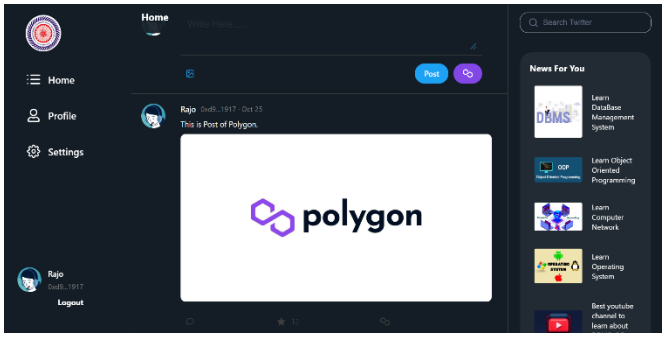

Figure – 10
Final Output and Generate the Hash

V. CONCLUSION AND FUTURE WORK

Creating immutable ledgers is one of the main values of Blockchain. This behavior helps us to achieve a system in which all the process is transparent and unchangeable. A block chain-based decentralised Daap application has been implemented in this work, which provides greater integrity and security than a traditional application. The proposed framework does away with the need for a centralised repository. This method decentralises the Dapp framework and eliminates the project's reliance on centralised computing resources for storage, processing, and uptime.

This decentralised technology based on block chains can also be applied to other social media applications such as Facebook, WhatsApp, and LinkedIn, among others. This, in turn, will aid in improving the Internet performance monitoring required to maintain the quality-of-service required for current and future Internet technologies.

In future we will make the Certificate Verification System that will help the Universities and Colleges (Shortens the verification time of the college degree 2-3 days to 10 minutes).

REFERENCES

[1]. Valeti Deepika, 2D. Lalitha Bhaskari 1M.Tech Student, Department of Information Technology, 2Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A), Andhra University, Visakhapatnam, AP, India. © 2020 JETIR November 2020, Volume 7, Issue 11 www.jetir.org (ISSN-2349-5162)

[2]. Md. Sabab Zulfiker, Nasrin Kabir, Al Amin Biswas, Partha Chakraborty and Md. Mahfujur Rahman, "Predicting Students' Performance of the Private Universities of Bangladesh using Machine Learning Approaches" International Journal of Advanced Computer Science and Applications(IJACSA), 11(3), 2020.

[3]. Dinesh Kumar K,Senthil P, and Manoj Kumar D.S, "Educational Certificate Verification System Using Blockchain", International Journal of Scientific & Technology Research, ISSN2277-8616,Vol.9, No. 3,pp.82-85, 2020

[4]. Hailong Yao, Caifen Wang, "A Novel Blockchain-Based Authentication Key Exchange Protocol and Its Applications," 2018 IEEE Third International Conference on Data Science in Cyberspace

[5]. Tareq Ahram, Arman Sargolzaei, Saman Sargolzaei, Jeff Daniels, Ben Amaba, "Blockchain Technology Innovations," 2017 IEEE Technology & Engineering Management Conference (TEMSCON)

[6]. "Getting Started," Getting Started - web3.js 1.0.0 documentation. [Online].
Available: https://web3js.readthedocs.io/en/v1.3.0/gettingstarted.html. [Accessed: 08-Mar-2021].

[7]. Ropsten Ethereum (rETH) Faucet. [Online]. Available:
https://faucet.dimensions.network/. [Accessed: 08-Mar-2021].

[8].MetaMask.[Online].
Available: https://metamask.io/index.html. [Accessed: 08-Mar-2021].

[9]. Welcome to Remix's documentation!," Remix. [Online]. Available: https://remix-ide.readthedocs.io/en/latest/. [Accessed: 08-Mar-2021].