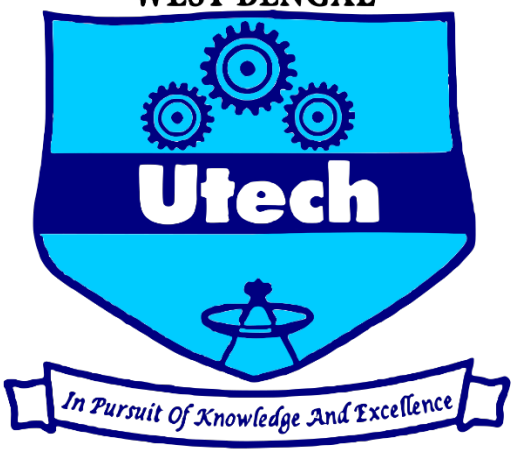
# Blockchain-based Data Storage System for Border Security Force

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL




# Maulana Abul Kalam Azad University of Technology, West Bengal

**(Department of Information Technology)**

---

Supervisor: Dr. Somdatta Chakravortty

Members and Contributors

Akash Halder, M.Tech IT, (2019-2021)
Chiranjeev Halder, MCA (2020-2022)
Koustav Kumar Mandal, M.Tech IT (IoT) (2019-2021)

# CONTENTS

# 1. ABSTRACT

These days, data leaks or breaches or unauthorized transmission of data within the current network is problematic. In some cases, intruders are exfiltrated data from the server several times without giving any clues. Databases are also vulnerable to attackers that are present inside the organizations. Insiders can also sometimes steal and tamper with the data, whether for money, profit or revenge. Our mission is to transform the centralized server system into a tamper-proof, consensus, time-stamping digital asset storing system. Blockchain is a distributed ledger technology and can be a software-based solution for managing and storing employee data. It is a chain of blocks linked with one another using cryptographic hash functions. It is a revolutionary peer-to-peer tree-like structure used to store digital assets to be tamper-proof, faster to access, and immutable. Considering the mentioned security issues of traditional data storage systems, we have designed the Ethereum blockchain-based data storage system for storing personnel record details.

# 2. INTRODUCTION

BSF is one of the largest armed forces in the world, guarding a total 6386.36 km India-Bangladesh, India-Pakistan border and Line of control area. Besides protecting the borders, BSF also earned a reputation for contributing to the country's internal challenges like election duty, counter-insurgency missions, disaster management, and more. BSF often plays vital roles in UN peacekeeping missions. The total strength of 257,363 trained men and women strives to protect and maintain the integrity of the country's border.

With the growing strength, it is essential to keep a personnel record system in the BSF. This record system plays a vital role in the management and is often used to calculate pay and manage workforce and personnel performance. Sometimes managing this system is a time-consuming and challenging task. Traditional record management systems are often unable to give the optimal data security and facing issues like data tampering, data leak and many data privacy issues.

Increasing rate of digital warfare and cyber-attacks protecting the record system is even more important. Network attackers are well known to that traditional record systems are vulnerable in nature and contains sensitive valuable information about the personnel. So traditional record keeping systems or databases are the primary soft target for the cybercriminals. There are so many numbers of key security failures that cybercriminals can take advantages.

Traditional database management system needs continuous maintenance and security tests after the deployment. Without proper maintenance sometimes traditional databases are behave abnormally and not doing things what the database is designed to do. Also, the maintenance and security tests result more costs.

1. In 2003, more than 75,000 computers were infected within 10 minutes of its deployment by a SQL worm because of Microsoft's SQL Server database vulnerability.
2. Databases also contain network interface and many times attackers try to capture network traffics. SSL- or TLS-encrypted communication platforms sometimes unable to protect metadata.
3. External attackers sometimes attack the system to steal data but many cybersecurity reports suggest, databases are vulnerable to attackers that are present inside at the organizations. Insiders are also sometimes can steal and tamper the data whether for money, profit or revenge. This the common problem presents in the modern organizations.
4. One of the critical threats for traditional SQL-based databases is SQL injections. Injections attack applications, and the database administrator are left to clean up the complexity caused by unclean variables and malicious code inserted into strings, later passed to an SQL server instance for parsing and execution.

Our developed blockchain-based personnel management system can overcome the current issues present in conventional databases. The system is a trust-based solution for personnel data managing and storing. The system stores cryptographic information in blocks and the blocks linked with other blocks using cryptographic hash functions.

The developed system uses blockchain to create a trusted environment to enhance data security, access control, and prevent attacks from internal or external sources. Blockchain is a revolutionary peer-to-peer tree-like structure capable of storing digital documents to be tamper-proof, faster to access, and immutable. The personnel data is secure and impossible to manipulate in the system because the network participants store and validate the system’s blockchain.

Every block in the chain contains a header and body. The header includes hash values of the preceding block and also present block and a nonce key. The block data is checked by the index method. Since the parameters of the earlier blocks influence the hash values recorded in each peer in the block, it is not possible to distort and modify the recorded data.

The transparency of the system is coming from the transaction coping process. All the transactions are copied to each node of the network. So, every participant can monitor each action performed by other members.

This report is prepared to outline the progress and in-depth details of work done so far of the project. We have created the structure of back-end and designed the front-end side of the system after facing several fundamental challenges throughout our development phase.

But we are glad to inform that one of the most difficult part of this project is completed.

## 3. OBJECTIVE

With the number of branch offices, the employee management system must record every present and past employee detail. The blockchain is suitable for maintaining an organized

record and data fetching also becomes quite efficient in various scenarios. Here comes the advantage of Blockchain technology, in the current centralized data storage system, a single data breach could compromise the data integrity and transparency of the entire network. Contrary to this, blockchain stores all sensitive data in a decentralized fashion, minimizing the probability of any cybersecurity threats. The data will also be available transparently, without authorizing anyone to omit or alter anything at any given instance.

Our primary objective is to

1. Build a decentralized database system based on a blockchain technology so that
    - no one can tamper with the original data.
    - the data is immune to unauthorized intrusion and data breaching.
2. In this system all the details of employees stored on the blockchain based decentralized storage system. And make a private blockchain network where all data transaction will happen.
3. Every block in the blockchain network decentralized across multiple trust worthy computers in the network making the model serverless.

Because of the serverless model single point server failure can be avoided, Serverless uses an event-based system versus stream-based. With event-based architecture, each subpart of the application is independent. Events trigger one another. In stream-based, there are connections to each service. If there is a failure, it just impacts that event, not the entire log. Being serverless, an organization is essentially outsourcing server and database management. Organizations are no longer responsible for the huge investments required for internal architecture administration. Ultimately organization's use case will define how much organization can save.

# 4. HARDWARE AND SOFTWARE REQUIREMENT

## 4.1 Hardware Requirements (For Development of 3 Nodes)

| Hardware | Configurations | Number of Items |
|---|---|---|
| **Computers powerful enough to handle the blockchain node setup minimum configuration as follows.** | 1. At least 16GB of RAM.<br>2. The processor at least INTEL i5 7th GEN processor.<br>3. At least 500 GB SSD | 3 Computers |
| **Router** | WIFI enabled | 3 |

## 4.2 Software requirements

| Backend | Frontend |
|---|---|
| Solidity | ReactJs |
| IPFS file server | JavaScript |
| Ganache (for development only) | MetaMask |
| NodeJS | Web browser |

## 4.3 Solidity

**Figure 1**

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state.

Solidity is a curly-bracket language. It is influenced by C++, Python and JavaScript, and is designed to target the Ethereum Virtual Machine (EVM).

Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

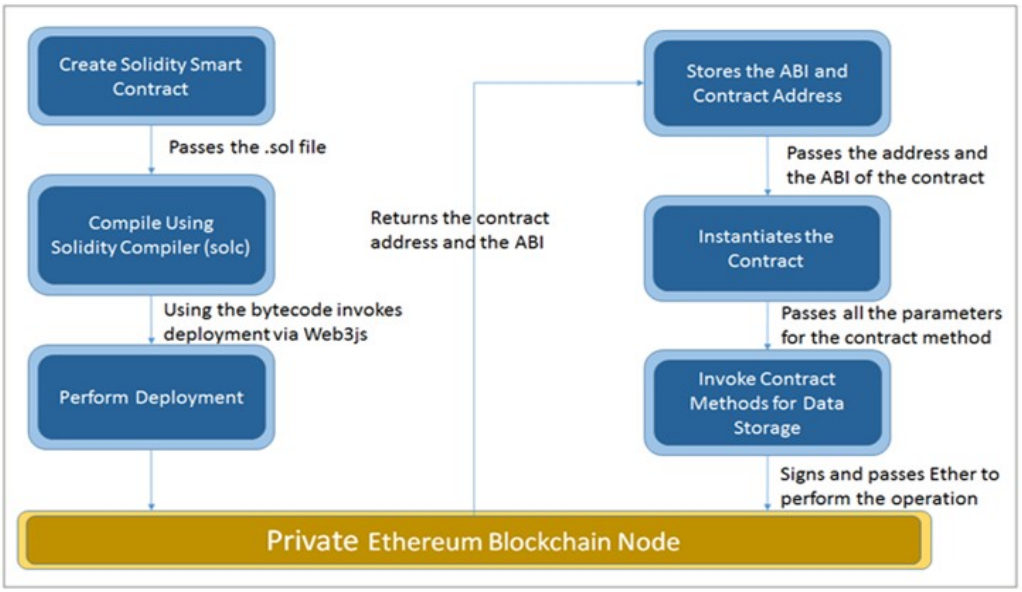



**Figure 2**

With Solidity programmers can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

## 4.4 IPFS (Interplanetary File System)

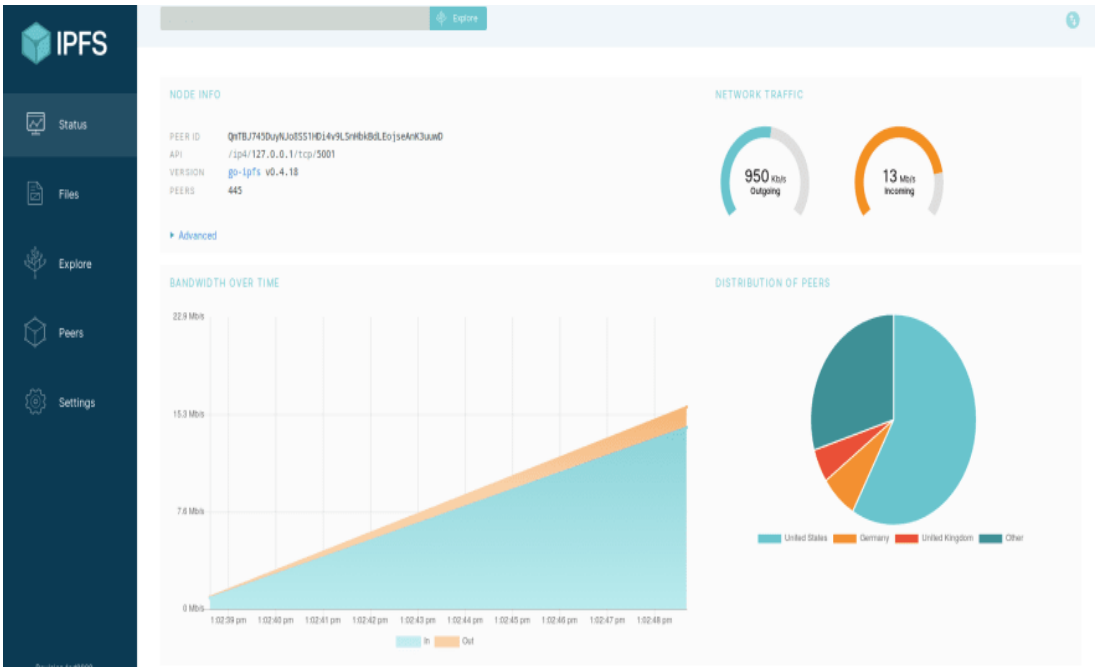



**Figure 3**

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.

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).

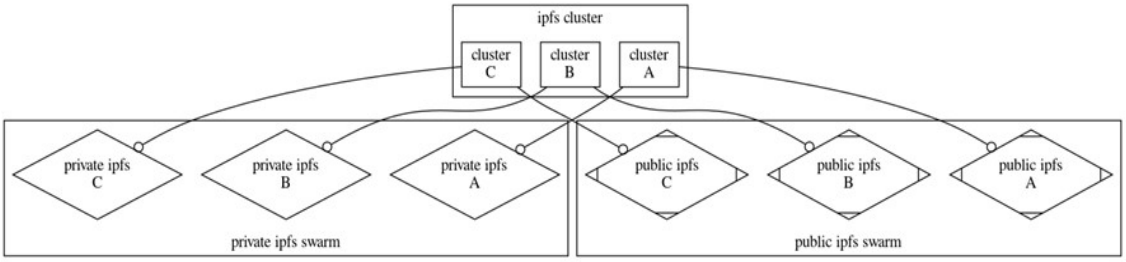



**Figure 4**

IPFS aims to create a single global network. This means that if Alice and Bob publish a block of data with the same hash, the peers downloading the content from Alice will exchange data with the ones downloading it from Bob. IPFS aims to replace protocols used for static webpage delivery by using gateways which are accessible with HTTP. Users may choose not to install an IPFS client on their device and instead use a public gateway. A list of these gateways is maintained on the IPFS GitHub page.

### 4.5 Ganache

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. You can use Ganache across the entire development cycle; enabling us to develop, deploy, and test our dApps in a safe and deterministic environment.

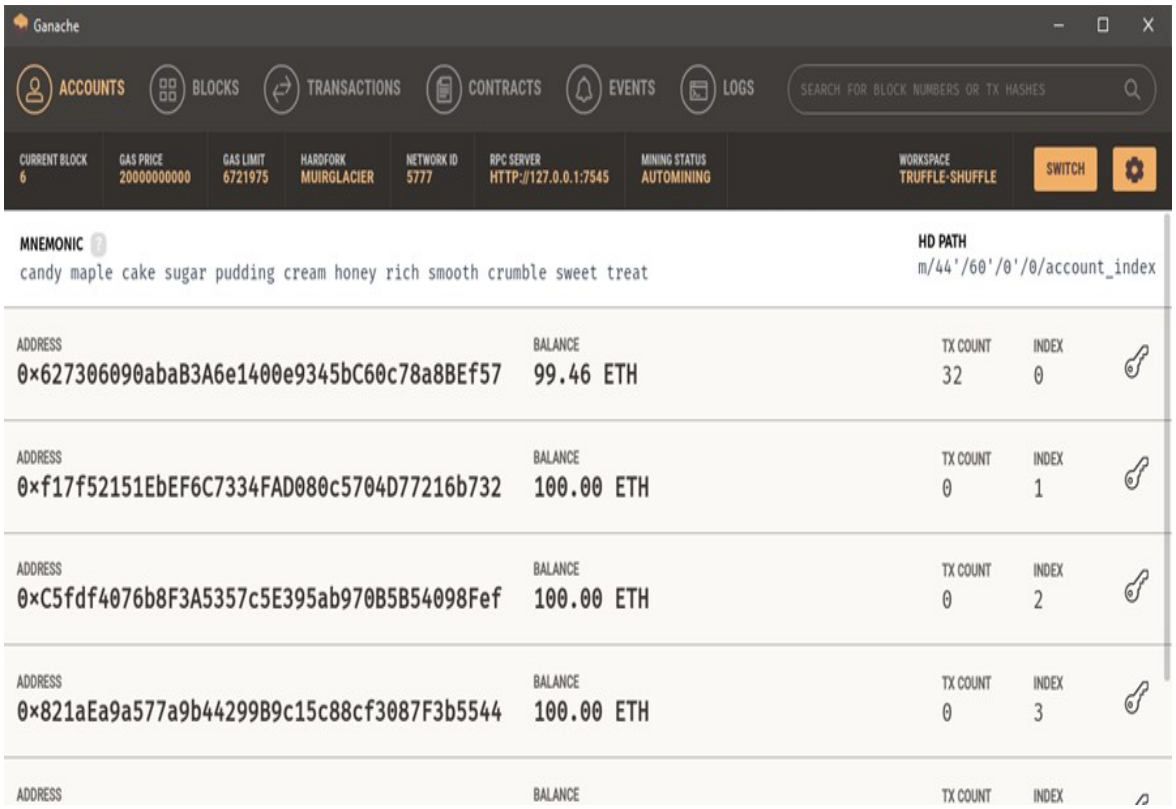



**Figure 5. Ganache**

Ganache comes in two flavors: a UI and CLI. Ganache UI is a desktop application supporting both Ethereum and Corda technology. The command-line tool, ganache-cli (formerly known as the TestRPC), is available for Ethereum development. Prefer using the command-line. This documentation will focus only on the UI flavor of Ganache.

### 4.6 NodeJS

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep. This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use. Furthermore, users of Node.js are free from worries of dead-locking the process, since there are no locks.

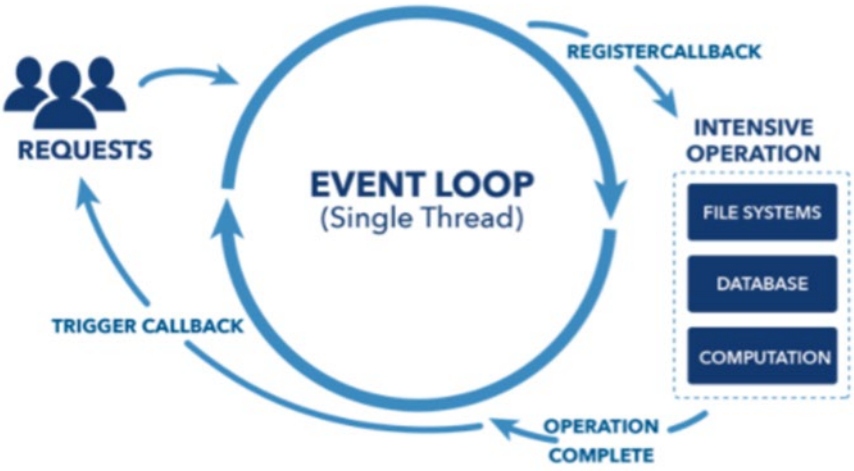


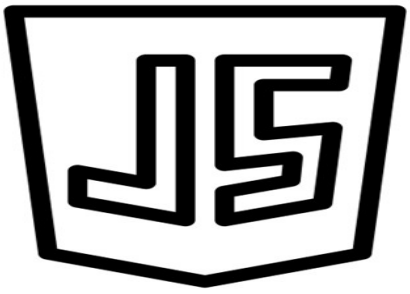

**Figure 6**

Almost no function in Node.js directly performs I/O, so the process never blocks except when the I/O is performed using synchronous methods of Node.js standard library. Because nothing blocks, scalable systems are very reasonable to develop in Node.js.

## 4.7 JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. Most web browsers have a dedicated JavaScript engine to execute the code on the user's device.



As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for

working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of other software systems, most notably servers and a variety of applications.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

### 4.8 ReactJs

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

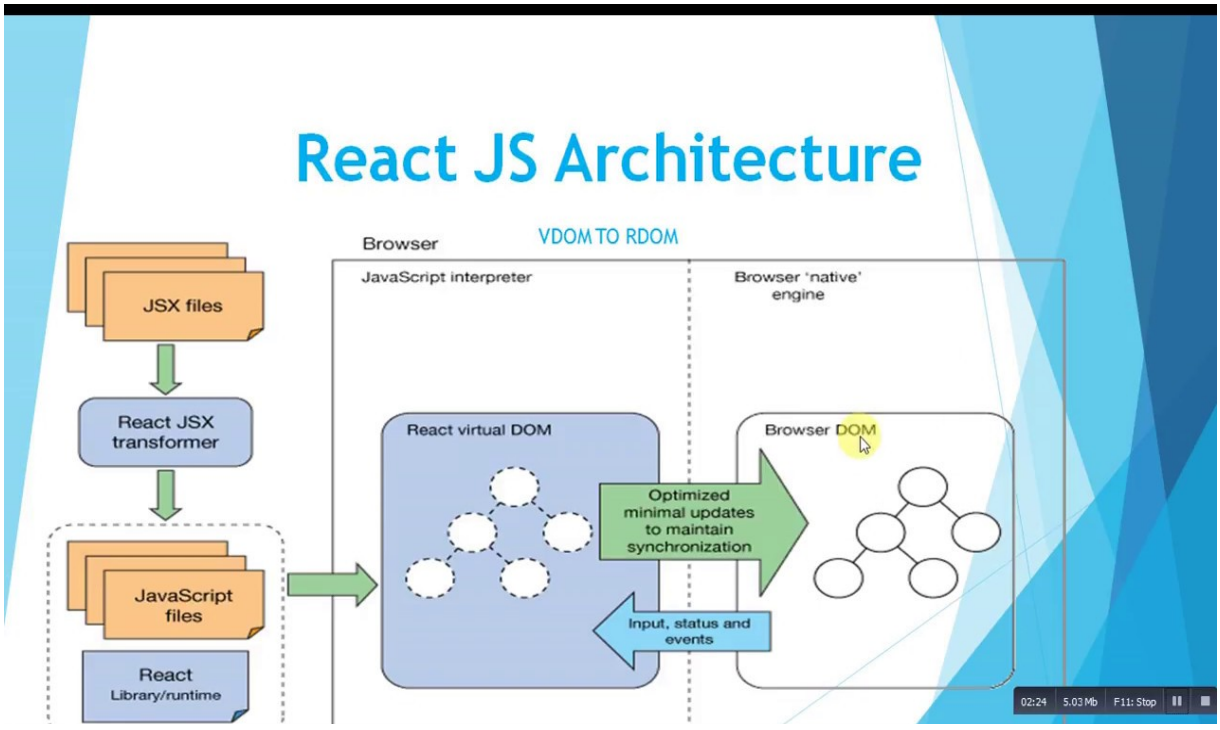



**Figure 7**

### 4.9 MetaMask

MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications. MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser.

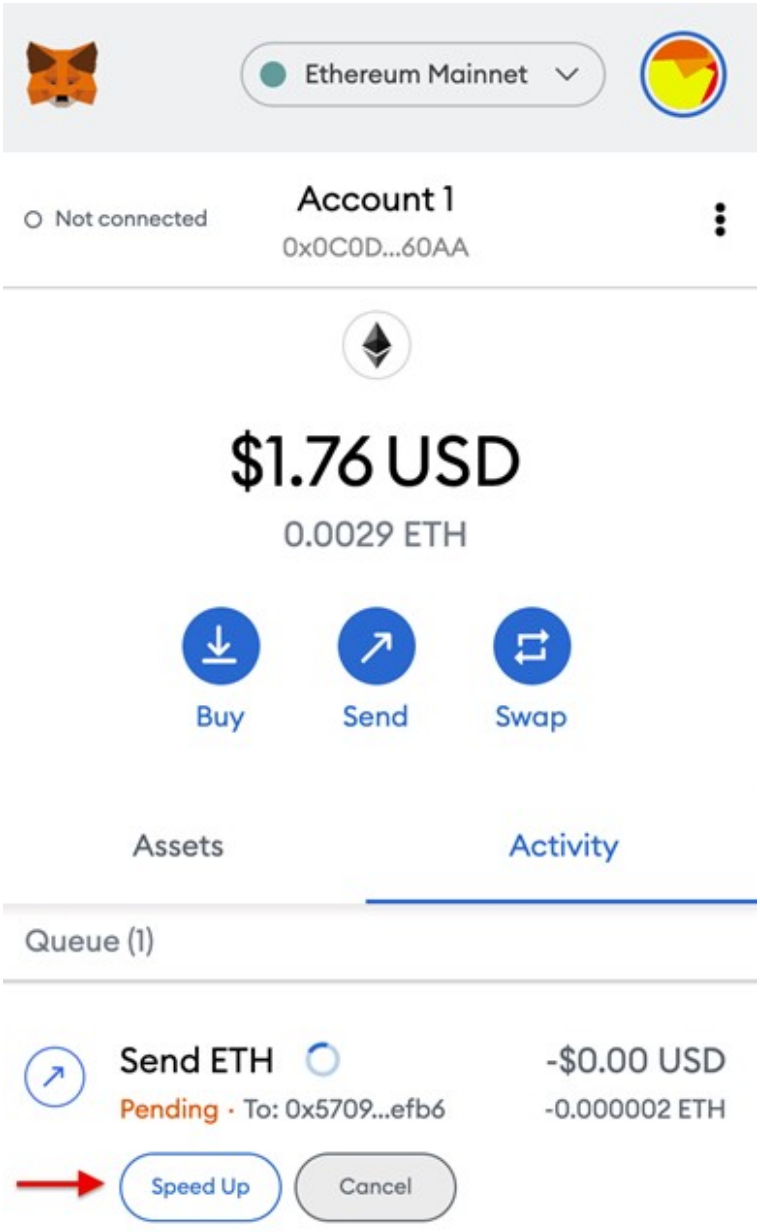



**Figure 8**

## 5. Methodology

The figure 10 shows the blockchain-based personnel management system and consists of two types of users

I. Administrator user
II. Authorized user

Every personnel record detail shown in the Table 1 is stored in the block, and the components can cooperate. Each user component is authenticated through the MetaMask software. After successful authentication, every user component can carry out its required data insertion and retrieval operations. The administrator remains at the highest level in the hierarchy. The administrator can insert new personnel record details and also update previous personnel details. The authorized user only can view the details of the personnel.

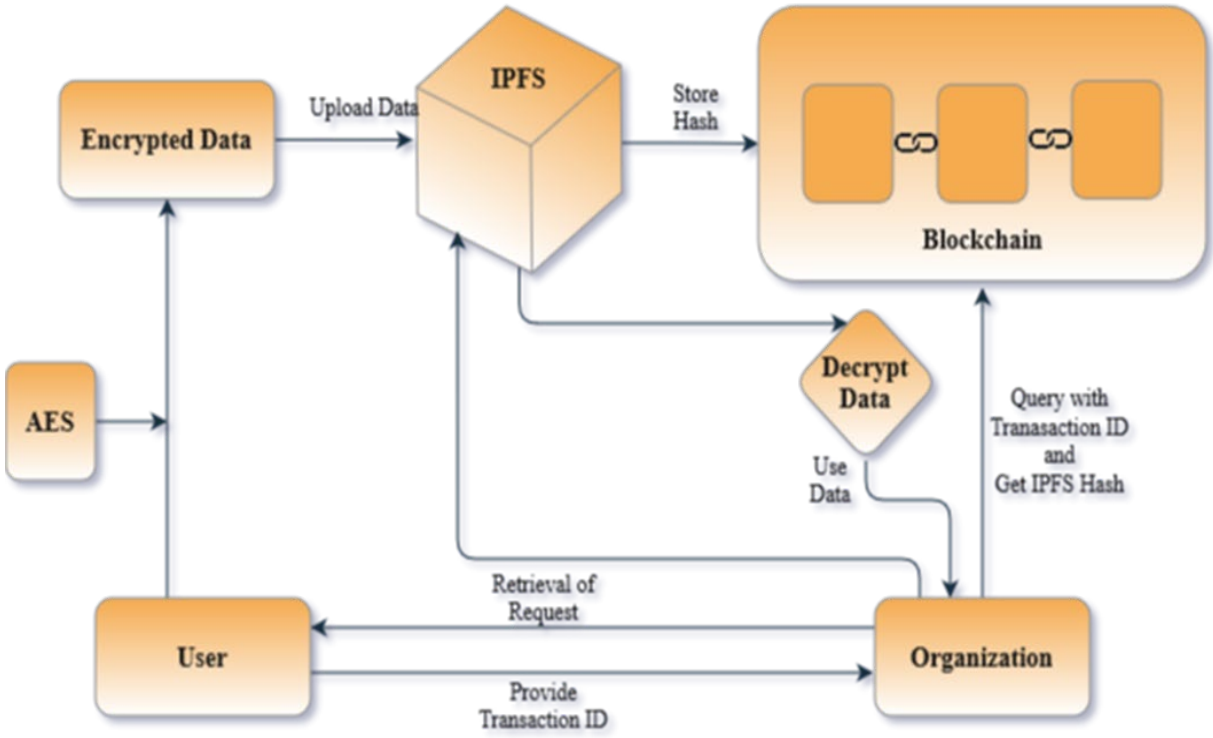



**Figure 9**

| Table 1. Personnel Record Fields | | |
|---|---|---|
| Sl. No. | **Field Name** | **Input Type of the field** |
| **1** | Name | String |
| **2** | IRLA Number, | Unique 9-digit Number |
| **3** | Seniority Number | Number |
| **4** | Rank | String |
| **5** | Unit | String |
| **6** | Parent Unit | String |
| **7** | Rank Held Since | Date |
| **8** | Appointment | String |
| **9** | Date Of Join In BSF | Date |
| **10** | Status In BSF | String |
| **11** | Email | Email |
| **12** | Date Of Birth | Date |
| **13** | Sex | String |
| **14** | Blood Group | String |
| **15** | Height | Number in cm |
| **16** | Fathers Name | String |
| **17** | Next To Kin | String |
| **18** | Nok Relation | String |
| **19** | Address | String |
| **20** | Religion | String |
| **21** | Category | String |
| **22** | Material Status | String |
| **23** | Home State | String |
| **24** | Family Details | String |
| **25** | Details Of Promotion | String |
| **26** | Educational Qualification | String |

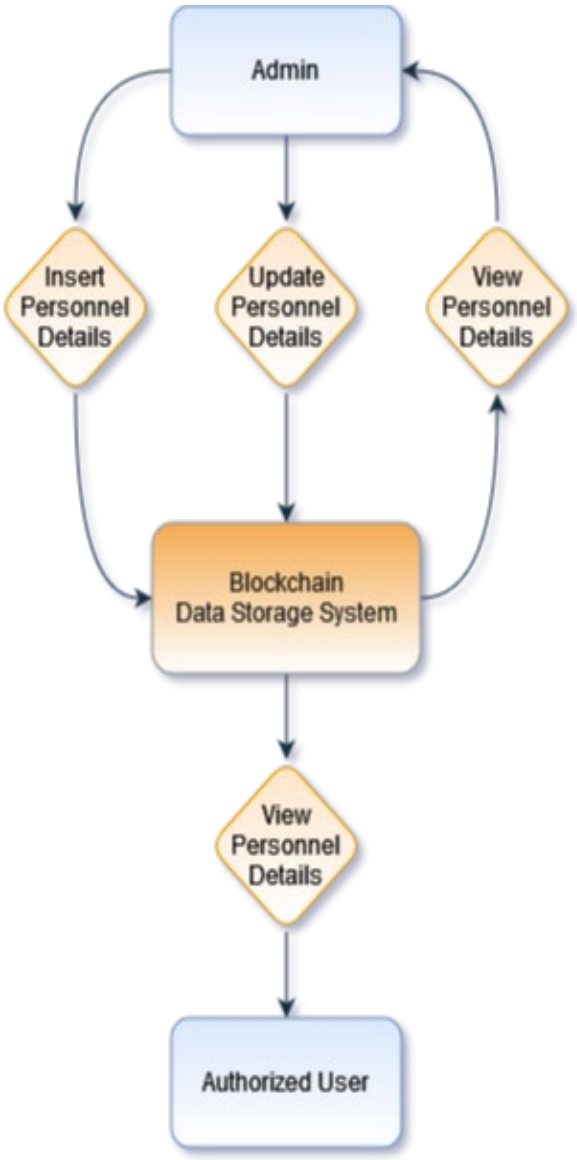



**Figure 10**

## 5.1 Date Storage

I. To Store Data on the Blockchain, ReactJS is used on the front-end which is used to take input from the user using form fields.

II. Advanced Encryption Standard (AES) methods of 256 bit are used to encrypt the input data taken from the user which provides one of the best encryption securities possible.

III. The Encrypted Data is sent to IPFS Client which distributes the data into chunks and generates a hash, which is required to find and decrypt the data on IPFS network.

III. Here MetaMask takes the private key of the user and checks if the user is authorized or not, if yes, then it sends the data storing request to the Blockchain network.

IV. The generated hash is sent to the Blockchain, Here Ganache is used to simulate a Blockchain network on a single computer for 'development purpose only'. The Blockchain again divides and distributes the data on the Blockchain network and keeps the hash and data both secured.

## 5.2 Data Retrieval

I. To retrieve the Data stored on the Blockchain, ReactJS is used on the front-end to take query from the user which includes a unique transactional ID and a private key. Here we used IRLA number which acts as a unique key in this distributed database.

II. Here MetaMask takes the private key of the user and checks if the user is authorized or not, then it sends the data retrieval request to the Blockchain.

III. The Blockchain network takes the private key and checks if it's authorized or not, and matches the IRLA Number against others and retrieves the IPFS hash which is used to store the details attached to that specific IRLA Number.

IV. With the IPFS hash the user sends it to the IPFS client, then it finds all the data related to it, and fetches the data to front end through server-side programming language NodeJS.

V. Once the data is ready, ReactJs on the front end takes the data and displays it in readable form to the user.

### 5.3 Modules

| Module Name | Description | Status |
|---|---|---|
| Back-End Structure | The sole structure of the system that stores and retrieves data to and from the Blockchain Network | Completed |
| Front-End Design | The user interface of the system that is used to interact with the user, that includes taking input from and showing output to the user | Completed |

## 6. Application Sample

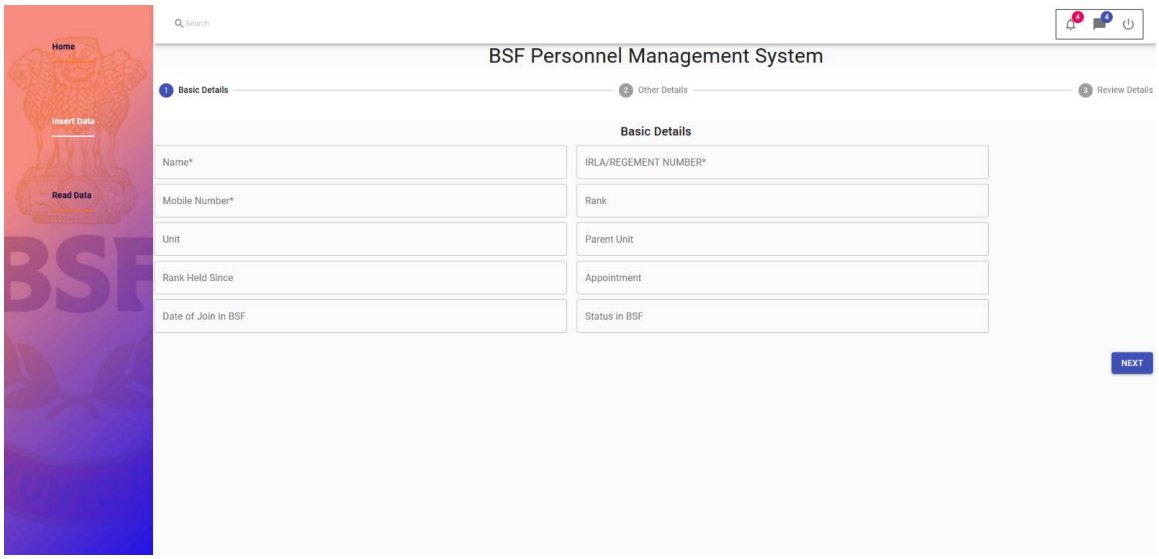



**Figure 11. Data insert Page**

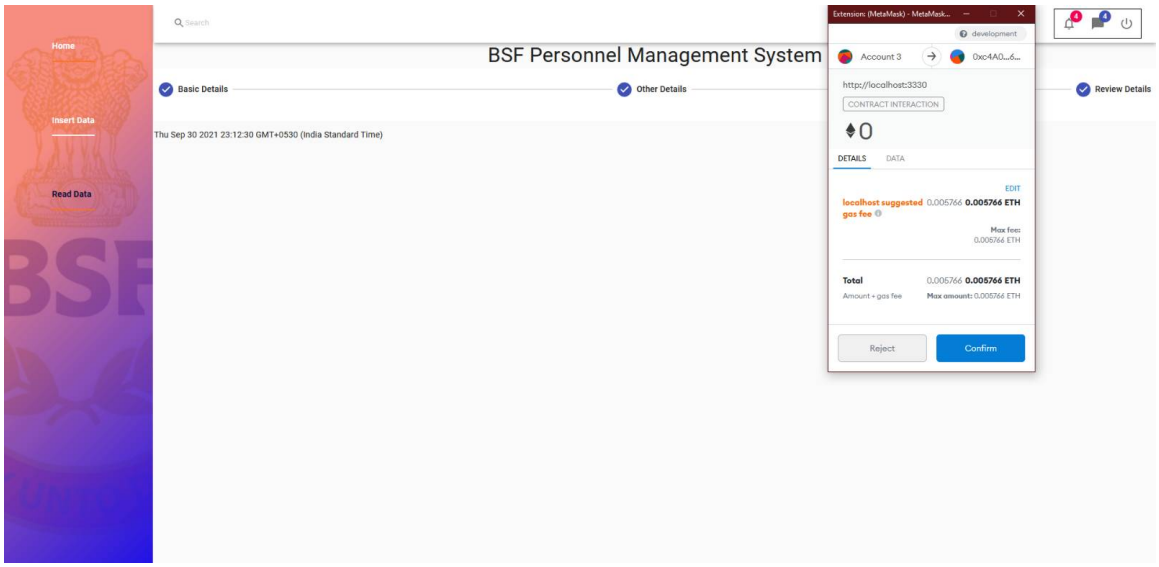



**Figure 12. Account Login Page**

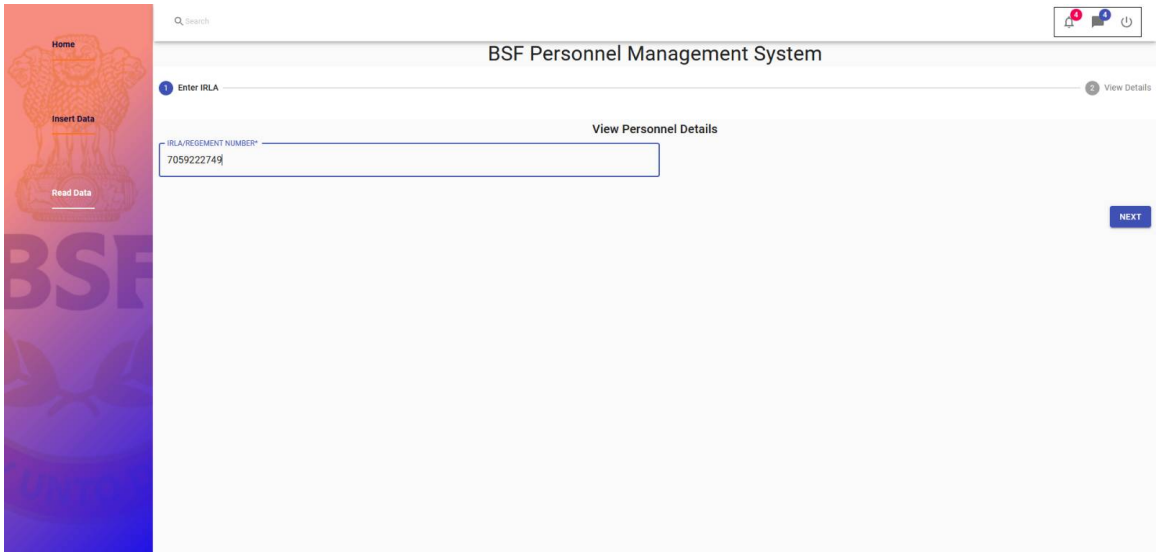



**Figure 13. Data Retrieval page**

## 7. Conclusion

In this project, we have developed the IPFS based data storage system for the border security force. The system has two primary users administrator and authorized user who participates in the blockchain network in a decentralized fashion to cooperate and perform the required operations of personnel records. The system is initially built using Ganache personal blockchain for prototyping purposes and storing and retrieving personnel details. In the future, we will further improve our blockchain-based system and utilize it for other types of transactions like file sharing, messaging, video, and audio on the blockchain.

# 8. References

[1] "Solidity — Solidity 0.8.10 documentation," Soliditylang.org. [Online]. Available: https://docs.soliditylang.org/en/latest/index.html. [Accessed: 01-Oct-2021].

[2] "Solidity & Ethereum in React (Next JS): The Complete Guide," Ebookee.com. [Online]. Available: https://ebookee.com/Solidity-Ethereum-in-React-Next-JS-The-Complete-Guide_4966192.html. [Accessed: 01-Oct-2021].

[3] Wikipedia contributors, "InterPlanetary File System," Wikipedia, The Free Encyclopedia, 01-Oct-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=InterPlanetary_File_System&oldid=1047471849. [Accessed: 01-Oct-2021].

[4] "Ganache," Trufflesuite.com. [Online]. Available: https://www.trufflesuite.com/docs/ganache/overview. [Accessed: 01-Oct-2021].

[5] "About," Nodejs.dev. [Online]. Available: https://nodejs.dev/about/. [Accessed: 01-Oct-2021].

[6] Wikipedia contributors, "JavaScript," Wikipedia, The Free Encyclopedia, 23-Sep-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=JavaScript&oldid=1046064964. [Accessed: 01-Oct-2021].

[7] Wikipedia contributors, "React (JavaScript library)," Wikipedia, The Free Encyclopedia, 28-Sep-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=React_(JavaScript_library)&oldid=1046983281. [Accessed: 01-Oct-2021].

[8] "About," Metamask.io. [Online]. Available: https://metamask.io/about. [Accessed: 01-Oct-2021].