

**TRANSPARENT GOVERNANCE:
EFFICIENT MONITORING OF
GOVERNMENT ALLOCATED FUNDS**

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

**PRINCE KUMAR KUSHWAHA (Reg. No - 40111001)
ANIMESH GUPTA (Reg.No - 40110092)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category -1 University by UGC**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

It is to certify that this Project Report is the bonafide work of Prince Kumar Jashwaha (40111001) and Animesh Gupta (40110092) who carried out the Project titled "Transparent Governance: Efficient Monitoring of Government Allocated Funds" under my supervision from November 2023 to April 2024.

Internal guide

Ms. R. YOGITHA M.E., (Ph.D.)

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY

Submitted for Viva voce Examination held on

13/04/2024

Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai - 600 075

Internal Examiner

External Examiner

DECLARATION

I, Prince Kumar Kushwaha (Reg.No- 40111001), hereby declare that the Project Report entitled "Transparent Governance: Efficient Monitoring of Government Allocated Funds" done by me under the guidance of Ms. R. YOGITHA M.E., (Ph.D.) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 13/4/2024

PLACE: Chennai

Prince Kumar

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. R. YOGITHA M.E., (Ph.D.,)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In an era where transparency, accountability, and the efficient use of public funds are paramount, the integration of blockchain technology into government fund allocation and tracking systems offers a transformative solution. This abstract provides a concise overview of a project that explores the implementation of blockchain for the management and oversight of government-allocated funds. The primary objective of this project is to harness the potential of blockchain technology to enhance the transparency, security, and efficiency of government fund allocation and tracking. By creating an immutable and decentralized ledger, we aim to ensure that public funds are allocated to projects and initiatives with maximum integrity. Our approach involves the development of a blockchain-based system that incorporates smart contracts to automate fund allocation and release based on predefined criteria and project milestones. Accessible via a user-friendly interface, the system employs cryptographic techniques to secure financial transactions and maintain a transparent ledger that is accessible to the public.

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vii
1	INTRODUCTION	1
	1.1 Traditional Fundraising Problem and Solution	1
	1.2 Purpose	2
2	LITERATURE SURVEY	4
3	REQUIREMENTS ANALYSIS	6
	3.1 Software Requirements Specification Document	6
4	DESCRIPTION OF PROPOSED SYSTEM	8
	4.1 Architecture / Overall Design of Proposed System	8
	4.2 User Interface	10
	4.3 Ethereum Smart Contract Design	11
	4.4 Implementation	12
5	IMPLEMENTATION DETAILS	14
	5.1 Front - End	14
	5.2 Back - End	14
	5.3 Algorithms	16
	5.3.1 AES Algorithm for Encryption	16
	5.3.2 MD5(Message-Digest Algorithm)	16
6	RESULTS AND DISCUSSION	17
7	CONCLUSION	22
	REFERENCES	23
	APPENDIX	26
	A. SOURCE CODE	26
	B. SCREENSHOTS	34
	C. RESEARCH PAPER	39

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO.
4.1	A Government allocated fund tracking Architecture	7
4.2	Library Component using Metamask to Communicate with Ethereum	8
4.3	Government Allocated Fund Tracker User Interface Design	9
4.4	Government Allocated Fund Tracker Smart Contract Design	10
5.1	Adding project name and new state page	14
5.2	Block Diagram of System	15
6.1	New User Registration	18
6.2	User registration successful	18
6.3	Sign up Page	19
6.4	Metamask Connection	19
6.5	Record's of Ethereum Transaction	20
B.1	Deployment Execution	34
B.2	Successful Deployment	34
B.3	Metamask Login	35
B.4	Metamask Connection Request	36
B.5	Metamask connection in thirdweb	37
B.6	Metamask Dashboard	38

CHAPTER 1

INTRODUCTION

N

Blockchain is characterized by its decentralized nature, the integrity of the information stored within the chain, and its openness. Due to these characteristics, another area in which Blockchain can be used is to release funds for government projects. Governments have to cater to an enormous number of responsibilities of a state. The working of state governments involves numerous transactions towards various operations that require to be applied throughout the state. This includes new projects, repair, and maintenance work, awarding contracts, paying off government employees, farmer schemes, and so on. A serious hurdle that the highest government faces is the low-level corruption that's sometimes impossible to trace, which deprives the state of progress. Tracking it's a really difficult task because of the present system. Blockchain is touted for its capability to reinforce the trust and ease of information-based exchanges among people and associations. The innovation offers a guarantee when deliberately applied within the proper settings.

India is among the largest democracies globally with 1.3 Billion populations, a major section of population is economically backward. The Government of India and the State Government's issues different policies and schemes on large scale for the economically lower-class population to benefit from it. At times the Centre and the State Governments issue some policies and schemes which most citizens are unaware of and the benefit of it is not availed by citizens.

There are cases where there are clashes in the State policies and Centre Government policies to overcome gap and to find and track the best scheme for citizens to avail benefit of these Government Schemes, Fund Tracker (State and Central) is be made using blockchain technique.

1.1 TRADITIONAL FUNDRAISING PROBLEM AND SOLUTION

In contemporary governance, the allocation and management of public funds pose substantial challenges. These challenges encompass the need for transparency,

accountability, and efficiency in disbursing taxpayer money for public projects and initiatives. Traditional methods often fall short in meeting these demands, leading to

issues such as corruption, misappropriation, inefficiency, and a lack of real-time oversight. Existing government fund allocation processes frequently lack transparency, making it difficult for citizens and stakeholders to access timely information about how public funds are allocated, disbursed, and utilized. Inadequate oversight and transparency can create opportunities for corruption and misappropriation of public funds, eroding public trust and damaging the credibility of government institutions.

The proposed solution leverages blockchain technology to revolutionize the way governments allocate and track public funds. By addressing the lack of transparency, inefficiency, and accountability in existing systems, this project seeks to instill trust, reduce the likelihood of financial misconduct, and optimize the effectiveness of public projects. The adoption of blockchain solutions tailored to government needs has the potential to usher in a new era of transparency and accountability in public fund management.

Blockchain offers the following potential solutions:

- Blockchain's immutable ledger ensures real-time public access to information about fund allocation and project progress, fostering transparency and trust.
- The integration of smart contracts can automate fund allocation and release, reducing bureaucratic overhead and expediting the flow of funds to projects based on predefined criteria and milestones.
- Blockchain's cryptographic techniques and decentralization enhance the security of financial transactions, mitigating the risk of fraud and corruption.

1.2 PURPOSE

The primary purpose of the project "Government Allocated Fund and Its Tracking Using Blockchain" is to address critical issues related to transparency, efficiency, accountability, and security in the management of public funds by government agencies. This project serves several significant purposes:

- **Enhancing Transparency:** The project aims to introduce a transparent and publicly accessible ledger for tracking government fund allocation and utilization. By utilizing blockchain technology, citizens and stakeholders can view and verify transactions in real-time, fostering trust and confidence in

government financial operations.

- Increasing Accountability: An essential purpose of this project is to establish mechanisms for holding government officials and project managers accountable for their decisions and actions regarding fund allocation. The immutable nature of blockchain records ensures that all transactions are verifiable, reducing opportunities for financial misconduct.

2 Streamlining Fund Allocation: By automating the fund allocation process through the use of smart contracts, this project seeks to streamline the bureaucratic and time-consuming procedures traditionally associated with government fund distribution

CHAPTER 2

LITERATURE SURVEY

[1] In this paper, the writer attempted to discover the ability of blockchain technology to decrease embezzlement which can also additionally arise for the duration of the fund's switch of government schemes. Also discovered 8 appropriate traits of any virtual fund switch method which incorporates a sequence of layers. To make certain equity on this form of the virtual method the writer has proposed to apply blockchain- based technology through the usage of mathematical model transaction traits to officially make certain equity at all levels. This system also can be audited through common human beings to trace the cash flow of any schemes, making the system absolutely obvious and fair.

[2] In this paper, the author considered blockchain applications that allow maintaining crystal clear records with the on-demand right to transactional data on a need-to-know basis. The system makes use of encryption to secure transactional data using hashes to take care of a block of transactions during a chain manner which is maintained and verified by every node involved to verify the transaction and save the info in an exceedingly transparent form within the government. In order to form a completely incorruptible government process, the system provides a way to allocate funds securely and track them authentically.

[3] This paper discusses the use of blockchain technology for tracking government funds. When the government issues funds for projects, often a large part of it is unutilized and is barely used for the actual project, one of the main causes being corruption. So, if blockchains are introduced here, everyone can track all the amounts regarding when and where it is being utilized. This technology is utilized in the government sector and can have a major impact on the growth and economy of a country.

[4] This system makes use of encryption to secure transactional data using hashes to take care of a block of transactions in a very chain manner which is maintained and verified by every node involved to verify the transaction and save the information in a very transparent way. Using the system, a transparent, incorruptible government process can be formed by providing a completely secure, authentic, and reliable system of allocation and tracking of funds.

[5] In this paper, the system provides tracking and allocation of funds at every single stage till it reaches the beneficiary's account. The system relies on hashing. The proposed framework expects to tackle all misrepresentations submitted within the public authority area and provides a sequence of secure exchange of information, cash, and other individual data of each transaction. There will be no requirement for the outsider and also the exchanges are regulated all the more sturdily and transparently. This system will assist in limiting human blunders and time delays in fund allocation.

[6] In this paper, System provides a decentralized application that provides the functionalities of a tracking system with proof of delivery using Quark Chain at its core. A transparent behavior of trade would be established based on smart contracts where the supplier and customer are on the same page and no third-party interference is needed. This would eliminate the scope of counterfeited products getting added to the supply chain as any inconsistencies in the ledger would be noticed, and necessary actions could be taken.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

3.1.1 Solidity

Solidity is an object oriented high-level language used for writing smart contracts. A statically typed language has features such as inheritance, libraries and complex user defined types. C++, python and JavaScript, majorly influenced it. Solidity is compiled to bytecode that is executable by Ethereum virtual machine (EVM). Using solidity developers can implement self-enforcing business logic contained in smart contracts, leaving an undeniable and permanent record of transactions.

3.1.2 Remix

Remix is an integrated development environment (IDE) used for developing smart contracts in solidity. Developing in remix assists developers to find bugs and debug code with ease. Remix supports three different kinds of environments to deploy and run the smart contract:

- Javascript VM: It creates a mock of blockchain environment so you can test your smart contract functionality.
- Injected Web3: This environment uses a browser plugin or a blockchain based browser such as Mist to connect to any Ethereum network (test or main).
- Web3 Provider: This environment connects to Ethereum node running at localhost and send the transactions to any network (test or main) as specified by the user.

3.1.3 Visual Studio

Visual studio code is a code editor developed by Microsoft. It is a feature rich editor that supports code highlighting, code debugging, intelligent code completion and code refactoring to build web applications. It supports wide range of programming

languages and has a built in terminal to execute command line commands. It is a

frontend, heavy application with lots of Javascript Visual studio code was a best choice as the editor to develop the application.

3.1.4 Metamask

Metamask is a chrome browser plugin that acts as a bridge between your browser and Ethereum blockchain by providing a secure identity vault, a user interface to manage multiple Ethereum wallets and sign blockchain transactions. It is one of the best ways to send transactions to Ethereum blockchain because it keeps a track of transaction execution and returns if any error occurs during mining or execution. It supports any ERC20 type token to be added to your wallet and trigger transaction on those ERC20 tokens. It is an Ethereum community open source project having more than million active users; hence, it is the most popular plugin to interact with blockchain.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The proposed system is used to track the funds granted to the state government as they go through the government process. It uses blockchain technology to safeguard transactions at each level while retaining transparency in every transaction and sealing every transaction with proof as the money goes forward. The system secures data using hashes to keep a block of transactions in a chain. It enables a complete proof, secure, and authentic financial distribution and tracking mechanism, which contributes to the formation of an incorruptible government.

In our system, there are 2 modules i.e., Admin (Government) and User.

Admin (Government) Module: Government provides the requested funds to the user.

User Module: In this system, the user will request the funds according to their needs and also, they can check their transaction history and wallet balance as well.

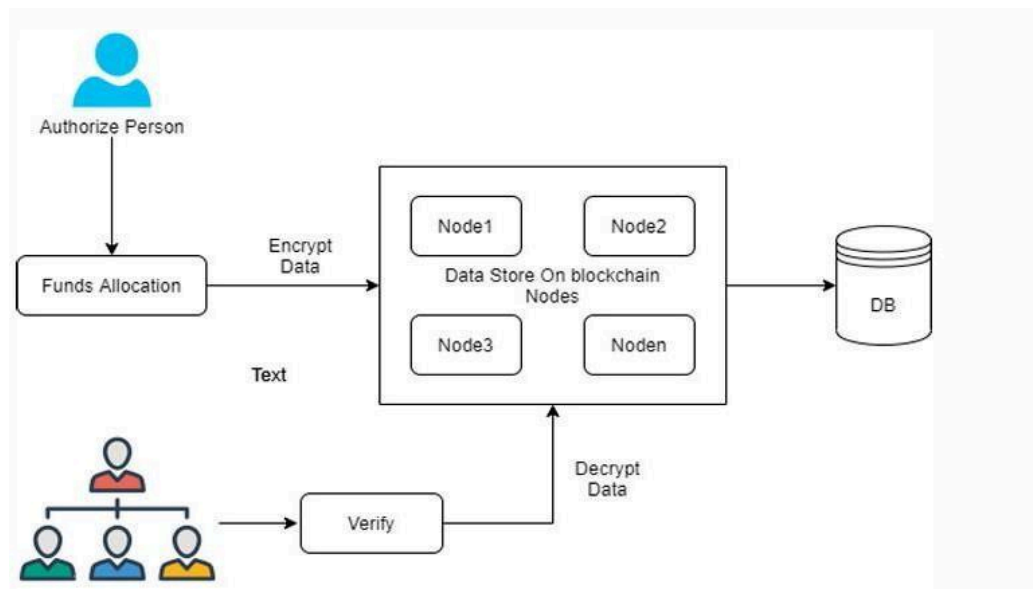


Fig 4.1: A Government allocated fund tracking Architecture

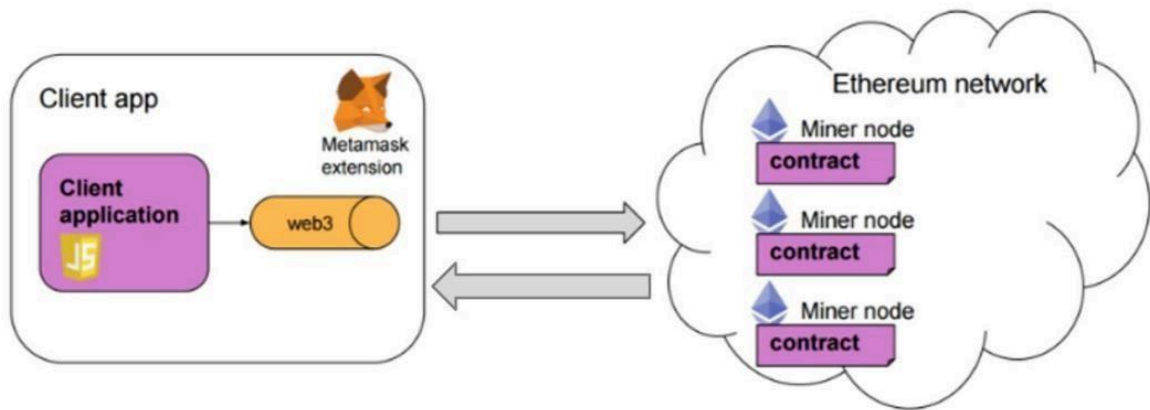


Fig 4.2: Library Component using Metamask to Communicate with Ethereum

As shown in Figure 4.2 library component using Metamask to communicate with Ethereum nodes library component uses web3 (Ethereum JavaScript API) to trigger transactions depending on the user interaction with the view component. Metamask acts as lightweight client that provides async functionality to library component for making calls to Ethereum network using API's provided by web3. Metamask also tracks the transaction processing and updates the library component about the status of mining of that transaction. A transaction needs to be signed before it can be sent to Ethereum blockchain. Metamask handles this by using the API's provided in web3 library. Once the transaction is signed Metamask will make a RPC call to send the transaction to Ethereum network where the smart contracts are deployed. Metamask polls Ethereum network to check the status of the transaction and update the Library component using a callback function present in library component.

The other network calls handled by library component is to the IPFS to store and retrieve campaign metadata. IPFS provides standard set of API's to interact with its file system. Library component makes web service calls to IPFS to store the metadata and receive a link to the stored data so in future it can just make a web call to retrieve that data.

The library component builds a data object with details received from IPFS and Ethereum network; it converts the data to a format suitable for view component to render in the browser. These helper functions take input in certain format such as big number and convert it to UI renderable strings. The view component becomes lightweight by moving all the data formatting logic to utility component. Library component mainly acts as the controller and service module as per the MVC architecture.

4.2 USER INTERFACE

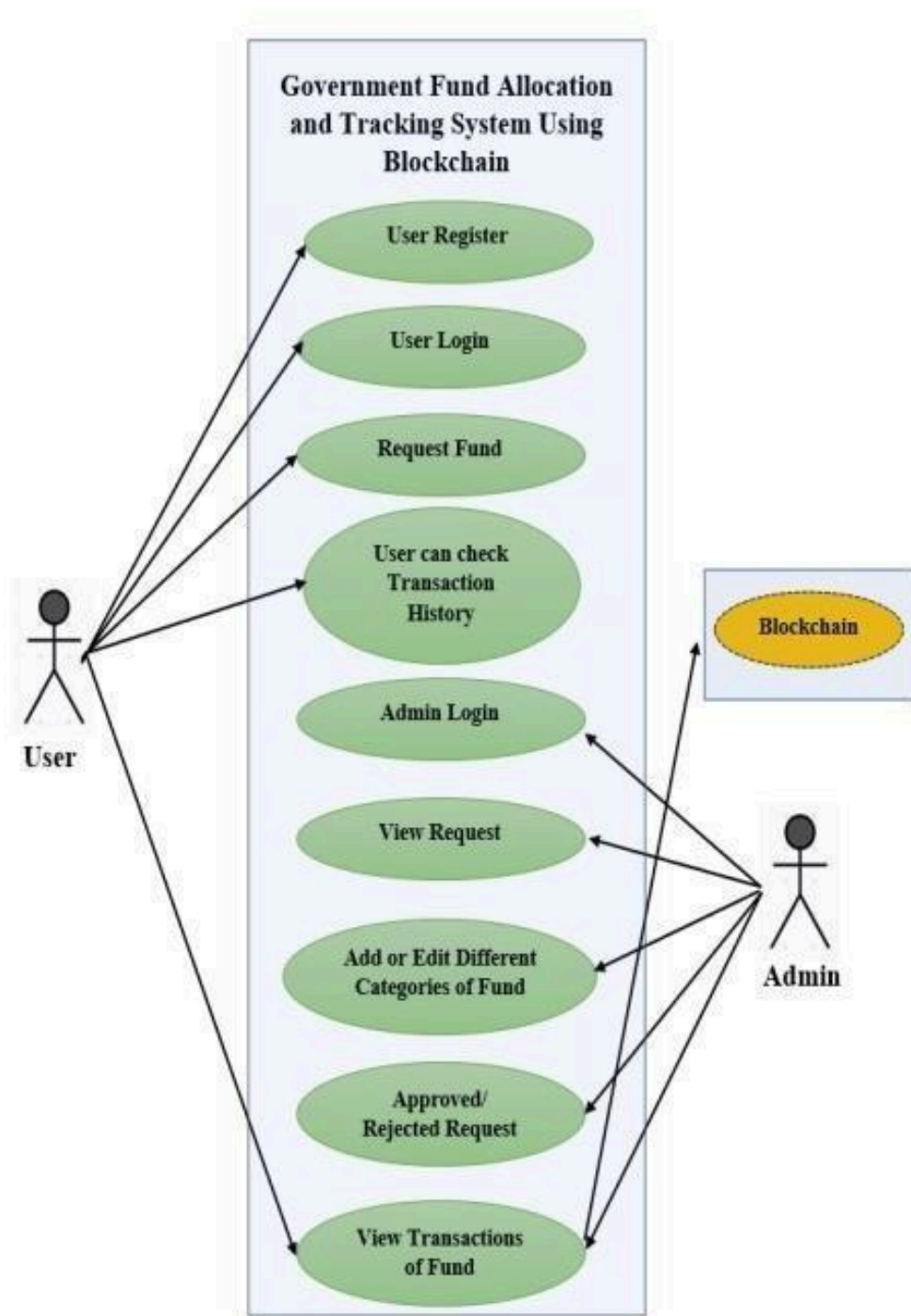


Fig 4.3: Government Allocated Fund Tracker User Interface Design

In the Figure 4.3 we can see that the user requests the funds from the admin (Government) then the request gets sent to the Government for approval. After that, the government views the request and then can approve or reject the request.

We can see that a block having the stated information is propagated into the network. The

transaction is validated by the network's nodes (people in real life). Following this confirmation, the block is put on the blockchain along with a timestamp. After that, the transaction could be enforced. All transactions submitted in this manner will be noted and made publicly available to everyone.

4.3 ETHEREUM SMART CONTRACT DESIGN

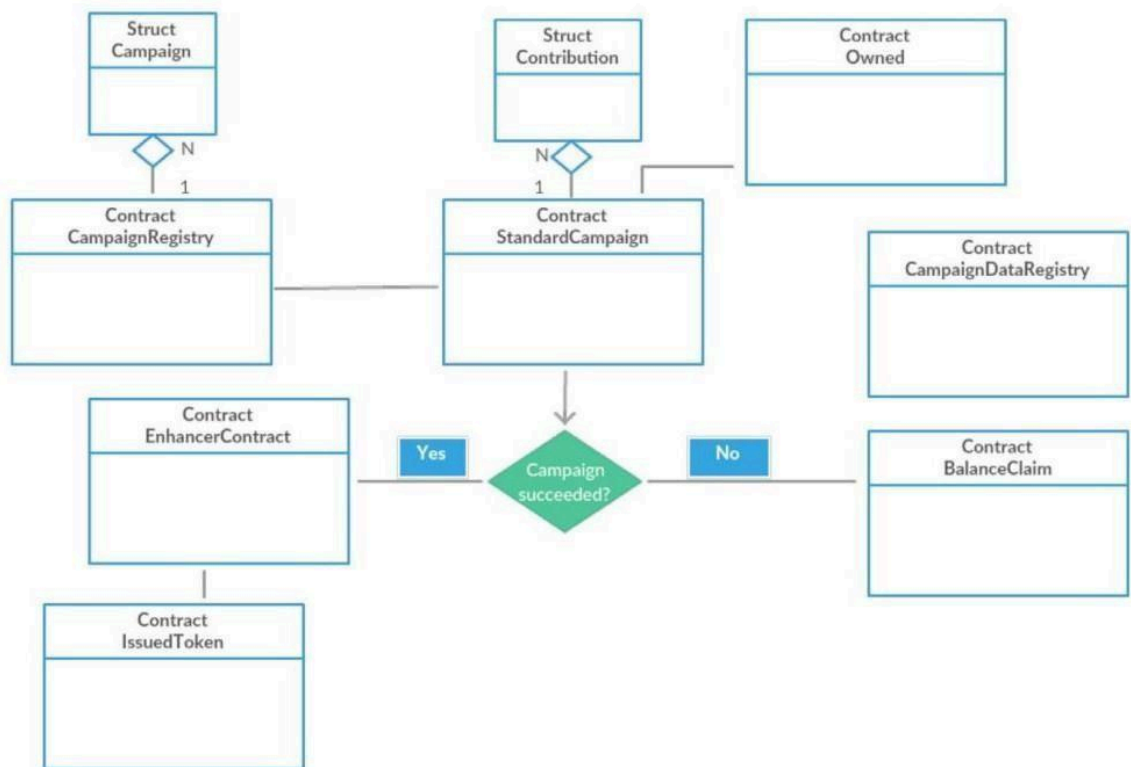


Fig 4.4: Government Allocated Fund Tracker Smart Contract Design

The root of the platform is Campaign Registry contract, which is created only once and contains all the campaigns reference that are created in the platform. When a new campaign contract is created it's contract address and contract creation timestamp is stored in the Campaign registry contract. It fetches all the contract references, stored in a struct datatype, from Ethereum network when the app is loaded and builds the landing page with campaign tiles.

The Campaign Data Registry contract contains information about IPFS hash, where a contracts metadata is present, so that when a campaign details page must be built the fund tracker can call this contract and use the reference returned to fetch a campaign's images, videos and other details. It consists of a map that takes in a campaign's address and returns the corresponding IPFS hash.

Issued Token contract contains details pertaining to a campaign specific token that is issued to the investor. It is associated with an enhancer contract that triggers the transfer of tokens to a particular user if the campaign succeeds.

Model Enhancer contract as the name suggests is an enhancer contract that contains details about each contribution to the campaign and calculates the number of campaign specific tokens to be issued in case the campaign succeeds. Standard Campaign contract is the most important contract that handles the business logic flow of a campaign. It contains the contributor contribution details and stage information of a campaign such as success, operational and failure details. Every campaign when created will have its own standard campaign contract that guides the execution of the campaign. Depending upon the result of funding standard campaign contract triggers model enhancer contract to disperse campaign specific tokens or creates a balance claim contract to transfer ether that has to be refunded to the investors. In case of a campaign failure standard campaign contract triggers multiple claim balance contracts for each investor to get his investment back. Standard campaign uses a struct data type called contribution to record all the contributions made to a campaign. It refers to this datatype in case of a campaign failure to create the balance claim contract created will have its own standard campaign contract that guides the execution of the campaign. Depending upon the result of funding standard campaign contract triggers model enhancer contract to disperse campaign specific tokens or creates a balance claim contract to transfer ether that has to be refunded to the investors. In case of a campaign failure standard campaign contract triggers multiple claim balance contracts for each investor to get his investment back. Standard campaign uses a struct data type called contribution to record all the contributions made to a campaign. It refers to this datatype in case of a campaign failure to create the balance claim contracts.

4.4 IMPLEMENTATION

The home page will have all the available schemes listed. A normal user can visit the site and check for schemes and available and allocated funds. A separate blockchain is available for each fund, where a user can check for all transactions. That transactions are publicly available. Users can check all the transactions within a particular block, all the information about that transaction is available here like

username, amount, date, and time of transaction. A new user can register to the site by following some simple steps and then the user can request funds. Admin login is also available. Once logged as admin there are options to add or edit any fund. Also, the admin can view all the pending transactions in a fund and can accept or reject any of the requests. A registered user can request funds and wait for the approval. The user wallet option is also available so the user can keep track of all his transactions and check the status of the amount available in the account and the approval status of requests. Admin can check for all the pending transactions requests and either can accept or reject them. Once the block is mined it cannot be modified.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 FRONT - END

We have developed a portal which provides the user to bid for the schemes that have been listed by the government for selling. For the front-end we have used web technologies like HTML5, CSS3 and JavaScript for adding the dynamicity to the portal and handling the logic part. This portal acts as the interface where the sellers (govt. Officials) and buyers (the ones bidding for particular schemes) interact and take the advantage of a full transparent and tamper proof system at their disposal.



Fig. 5.1 Adding project name and new state page

5.2 BACK - END

The backend of our project is fully developed using frameworks based on Blockchain technology like Bitcoin, Truffle suite. Ethereum is actually a personal Ethereum block chain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates. It provides you 10 ethereum accounts with 100ETH each for testing the network. Also each and every transactions taking place in the network is stored in blocks with some details like Timestamp, hash of the block, and other necessary details required to make the ledger complete. Truffle is actually a development environment, testing framework all in one. It is based on Ethereum and allows the smooth and seamless development of DApps i.e. Distributed Applications. With truffle you can compile the smart contracts and deploy them into web applications and using it you can develop the frontend for your applications. The smart contracts are developed using the Solidity language that we have used to set the terms of agreement

between the buyer and sellers of the schemes. For the payment gateway we have used the MetaMask which acts as a wallet for your web applications. All the transactions taking place are confirmed using the metamask.

PROCESS: Including many projects, maintenance or repairing work and recording employment of the area and many more are included in the role activity of the state as well as central government. A major hurdle that the top government faces is the low level corruption that is sometimes impossible to track which deprives the state of progress. Hence we are going to propose a theory based on the blockchain, which includes the tracking of every other transaction done within the city on the basis of small scale industries or large scale companies.

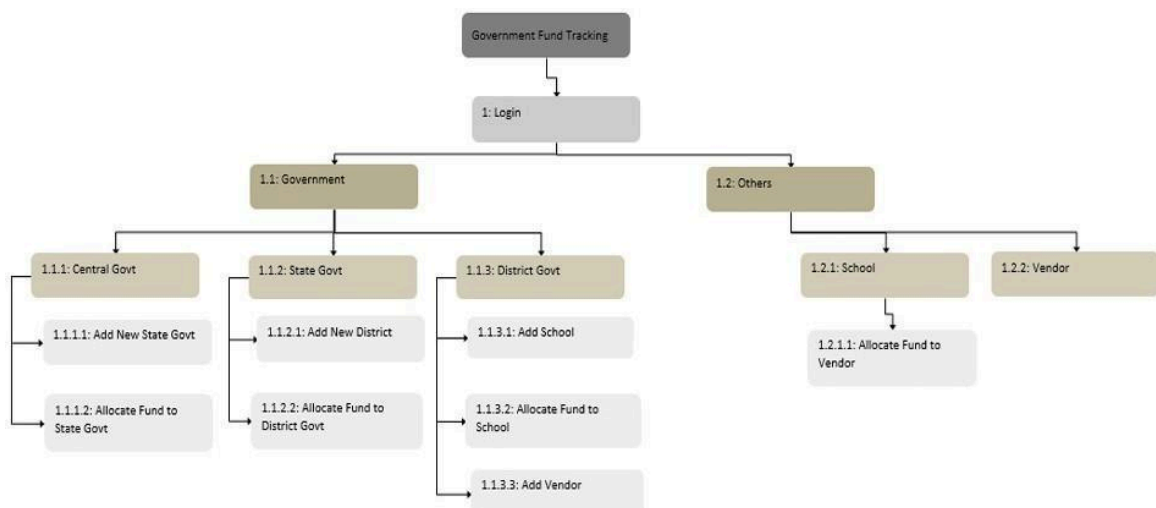


Fig. 5.2 Block Diagram of System

This allows to maintain a crystal clear record with on-demand right to transactional data on a need to know. For the payment gateway we have used the MetaMask which acts as a wallet for your web applications. All the transactions taking place are confirmed using the metamask.

5.3 ALGORITHMS

5.3.1 AES Algorithm for Encryption

AES (advanced encryption standard). It is symmetric algorithm. It used to convert plain text into cipher text. The need for coming with this algorithm is weakness in DES. The 56 bit key of des is no longer safe against attacks based on exhaustive key searches and 64-bit block also consider as weak. AES was to be used 128-bit block with 128-bit keys.

Rijndael was founder. In this drop we are using it to encrypt the data owner file.

Input:

128_bit /192 bit/256 bit input (0, 1)

Secret key (128_bit) +plain text

(128_bit). Process:

10/12/14-rounds for-128_bit /192 bit/256 bit input

Xor state block (i/p)

Final round: 10, 12, 14

Each round consists: sub byte, shift byte, mix columns, add round key. Output:

Cipher text (128 bit)

5.3.2 MD5(Message-Digest Algorithm)

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

Steps:

- A message digest algorithm is a hash function that takes a bit sequence of any length and produces a bit sequence of a fixed small length.
- The output of a message digest is considered as a digital signature of the input data.
- MD5 is a message digest algorithm producing 128 bits of data.
- It uses constants derived to trigonometric Sine function.
- It loops through the original message in blocks of 512 bits, with 4 rounds of operations for each block, and 16 operations in each round.
- Most modern programming languages provides MD5 algorithm as built-in functions

CHAPTER 6

RESULTS AND DISCUSSION

Blockchain technology is revolutionary. It will make life simpler and safer by changing the way personal information is stored and how transactions for goods and services are made. By keeping this in mind, we have proposed a solution in blockchain technology that helps us generate a record of all the transactions such that each transaction is saved as a connected block. We have used technologies such as hashing, graph DB, platforms, etc. that have helped us develop a systematic approach to track all the records while considering security as a priority. We have considered that in blockchain technology there are some access and privacy challenges. Though, it allows us to maintain a crystal clear record with on-demand access to the right transactional data on a need-to-know basis. In the future, it may be possible that we don't need to rely on a third party in transactions. As we know prominent organizations in the world today function as trusted third parties, for instance, SWIFT and the Depository Trust Cleaning Company. Corporate chances flourish for companies that build and apply Blockchain technologies aimed at particular transactions, like the mortgage industry.

Benefits of this system:

Transparency: The blockchain ledger provides real-time, transparent tracking of all government allocation transactions. Citizens, government officials, and auditors can view and verify transactions, promoting trust and accountability.

Security: Blockchain's inherent security mechanisms make it highly resistant to fraud and unauthorized alterations of data. This enhances the integrity of government fund management.

Efficiency: Automated smart contracts can expedite the allocation process, reducing administrative overhead and the time required for fund disbursement.

Accountability: With transparent records and automated processes, it becomes easier to hold government officials and agencies accountable for their financial decisions

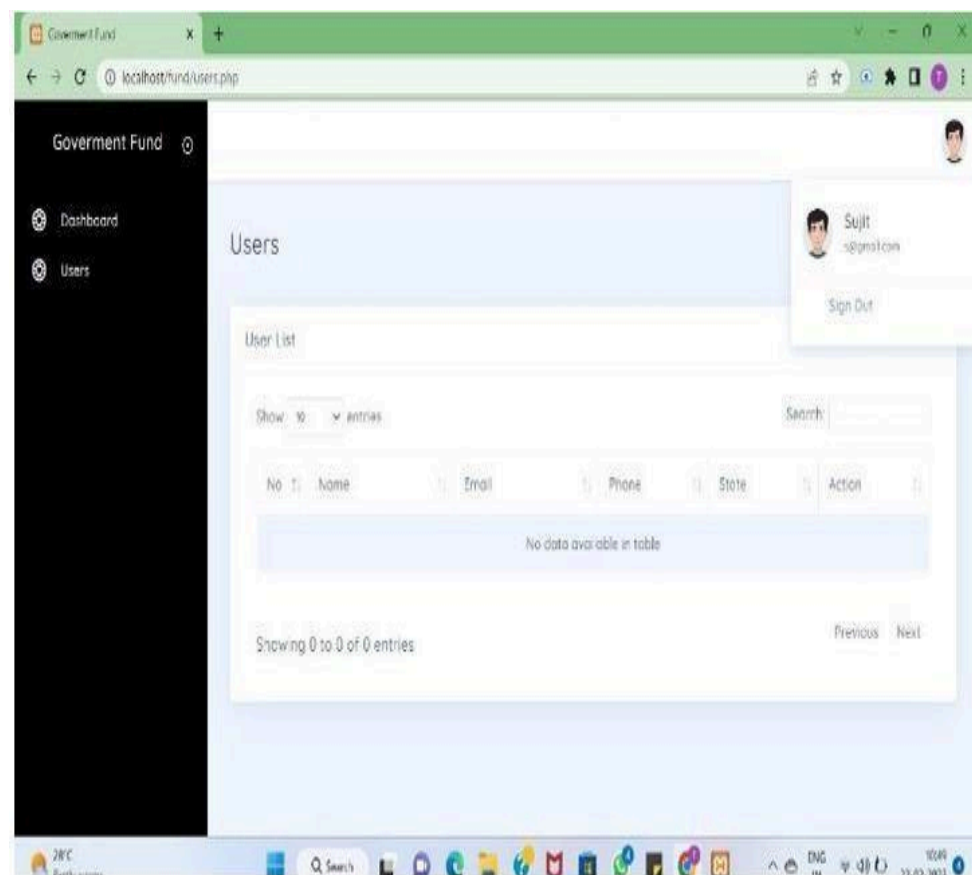
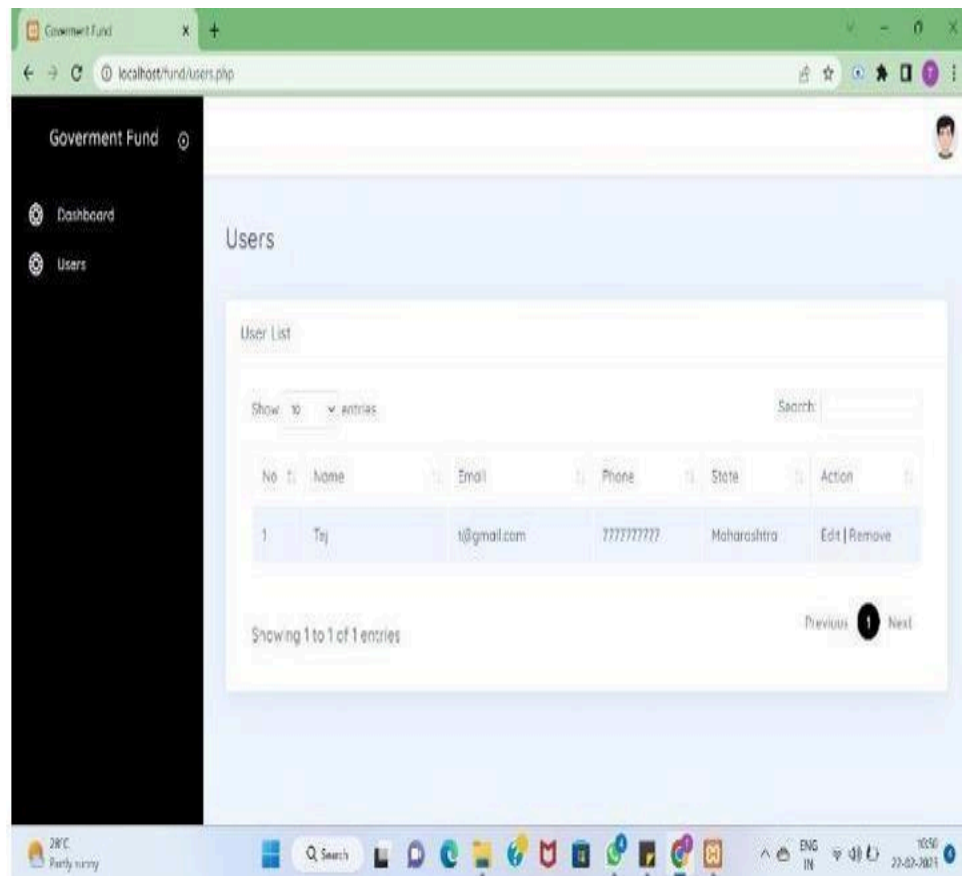


Fig. 6.1 New user registration

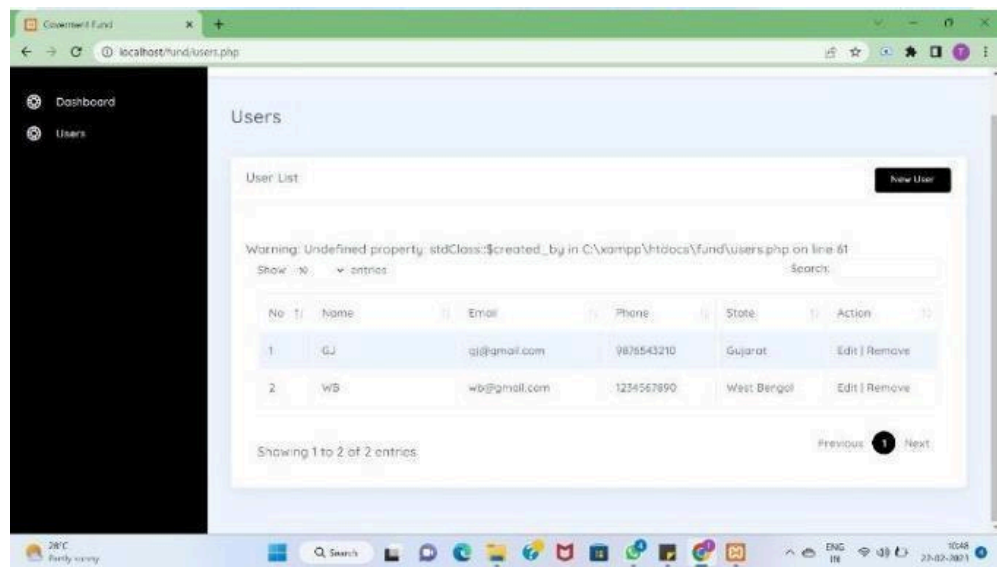


Fig. 6.2 User registration successful

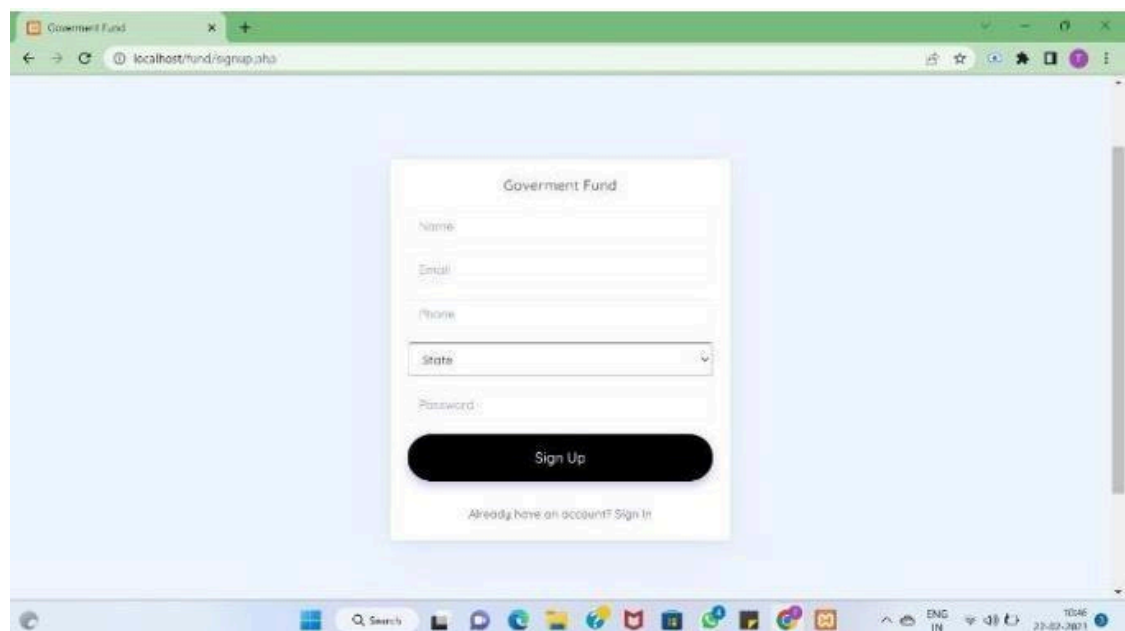


Fig. 6.3 Sign up Page

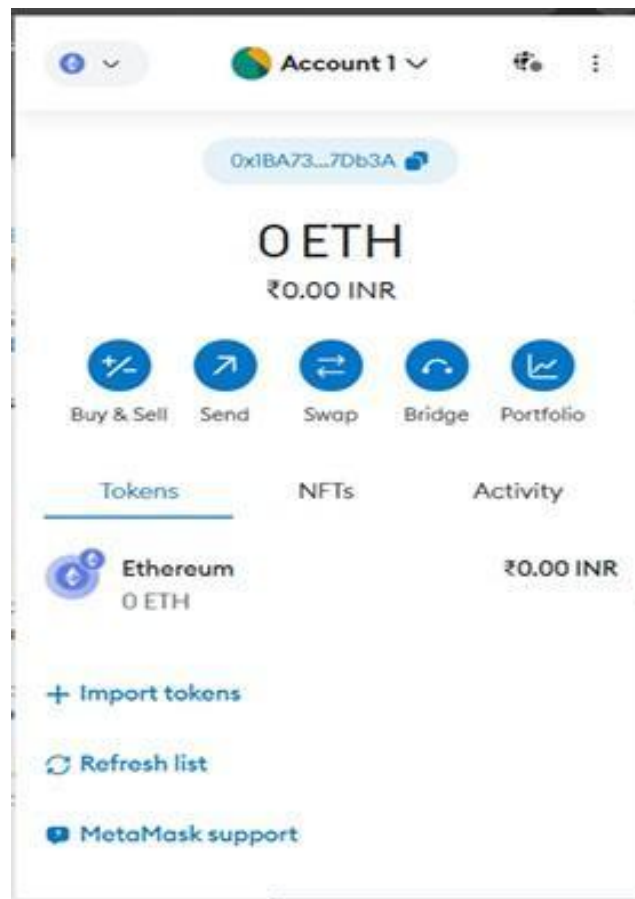


Fig. 6.4 Metamask connection

Transactions						
Erc20 Token Trns						
Latest 6 trns						
TxHash	Block	Age	From	To	Value	(Info)
0xe1290e614201ca1...	3034353	14 days 16 hrs ago	0x5869d3c7bb63230...	OUT 0x187235a3c4501ba	0.3 Ether	0.00047295
0x90a21e41ca0eb4...	2947725	40 days 5 hrs ago	0x5869d3c7bb63230...	OUT 0x0c1eaa43c3c1259...	0 Ether	0.000002342
0xf52ac5a6f5a81e8...	2766421	72 days 6 hrs ago	0x21638d1c0073dee...	IN 0x5869d3c7bb63230...	7.5 Ether	0.00002

Fig. 6.5 Record's of Ethereum Transaction

FUTURE SCOPES

- The system can be made more scalable by using the Byzantine consensus mechanism.
- The system can be further secured by adding another level of encryption to data.
- The usage of multiple documents can be bundled together to use one unique identifier which will make tracking easier.
- Sub-contractors can be added so that funds for schemes can go directly to them whenever necessary instead of citizens having to find contractor and sub-contractors.

CONCLUSION

The features of Blockchain like immutability, tamper-proof, secured and decentralized approach helps in eliminating the security vulnerability of the application. Hyperledger fabric is like other blockchain technologies which has a ledger, uses smart contracts and is a system by which participants manage their transactions. It provides proper governance and access control and can be scaled up whenever needed. We have to consider about the blockchain applications, we even have to consider the access and privacy challenges though. This allows to maintain crystal clear record with on demand right to transactional data on a need-to-know basis. The system makes use of encryption to secure transactional data using hashes to maintain a block of transactions in a chain manner which is maintained and verified by every node involved to verify the transaction and save the data in a transparent form within the government. The system allows for a full proof, secure and authentic fund allocation and fund tracking system to help form an incorruptible government process. Even then, with further enhancements, this blockchain model can provide a transparency in all the government transactions. There will be no discrepancies of any kind. Because of the decentralized ledger all the transactions can be verified and cannot be altered. This proposed framework is added to assist the authorities to lessen corruption and offer transparency in all transactions because of the functions of blockchain-like immutability, proof of work, and security. It offers the right governance and transparency. It will maintain track of all transactions made. As blockchain technology is used the transactions as soon as made cannot be changed and if there's any try of tempering, we can get to recognize approximately that easily. There might be no requirement for the outsider and the exchanges might be regulated all the extra sturdily and transparently. In addition to preventing human errors and delays, it will help eliminate human errors. This framework will make the general public authority framework activities appreciably extra stable and productive. The money that is released can be tracked, anyone and everyone can find out how the money is being used. Such a blockchain will surely reduce the ongoing corruption. It will create a huge impact on the economic development of a country.

REFERENCES:-

- [1] P. Joshi, S. Kumar, D. Kumar, and A. K. Singh, (2019) "A Blockchain-Based Framework for Fraud Detection", Conference on Next Generation Computing Applications (Next Comp).
- [2] Sahil Siddharth Jambhulkar, Vishakha Prashant Ratnaparkhi (2020) "Government Fund Distribution and Tracking System Using Blockchain Technology", International Journal of Emerging Technologies and Innovative Research, ISSN:2349-5162, Vol.7, Issue 9, page no.1379-1387.
- [3] A. Mohite and A. Acharya, (2018) "Blockchain for government fund tracking using Hyperledger", International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS).
- [4] Abhishek Katore, Sanskar Choubey, (2021) "Government Scheme and Funds Tracker using Blockchain", International Journal of Engineering Research and Technology (IJERT), Volume 10, Issue 05.
- [5] Meghna Vadher Shivani Pandey, Darshana Sawant, Hezal Lopes (2021) "State Government Fund Allocation and Tracking System using Blockchain Technology", International Journal of Emerging Technologies and Innovative Research ISSN:2349-5162, Vol.8, Issue 6, page no. a455-a459
- [6] . A Chauhan, G. Savner, P. Venkatesh, V. Patil and W. Wu, (2020) "A Blockchain-Based Tracking System", IEEE International Conference on Service-Oriented System Engineering (SOSE), Oxford, United Kingdom.
- [7] V. Buterin, "Ethereum Whitepaper," 2013. [Online]. Available: <https://github.com/ethereum/wiki/wiki/White-Paper>. [Accessed 21 September 2017].

- [8] Solidity. [Online]. Available: <https://solidity.readthedocs.io/en/develop/>.
- [9] ChristophM, "Intoduction to the DAO," [Online].
Available:
<https://daowiki.atlassian.net/wiki/spaces/DAO/pages/5996662/Introduction+to+the+DAO>.
- [10] "Attack," [Online].
Available:
<https://daowiki.atlassian.net/wiki/spaces/DAO/pages/7209155/Attack>.
- [11] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/Solidity>.
- [12] "Remix, Ethereum-IDE," [Online].
Available: https://remix.readthedocs.io/en/latest/run_tab.html.
- [13] "Wikipedia," [Online].
Available: https://en.wikipedia.org/wiki/Visual_Studio_Code.
- [14] G. Hayes, "The Beginners Guide to Using an Ethereum Test Network," 16 February 2018. [Online]. Available:
<https://medium.com/compound-finance/the-beginners-guide-to-using-an-ethereum-test-network-95bbbc85fc1d>. [Accessed 26 March 2018].
- [15] "Wikipedia," [Online].
Available: https://en.wikipedia.org/wiki/InterPlanetary_File_System.
- [16] L. Mearian, "Ethereum explores a fix for blockchain's performance problem," 5 January 2018. [Online].
Available: <https://www.computerworld.com/article/3245928/emerging-technology/ethereum-explores-a-fix-for-blockchains-performance-problem.html>.
- [17] A. Lielacher, "The Top Five Ethereum Competitors," 21 September 2018.
[Online]. Available:

<https://btcmanager.com/the-top-five-ethereum-competitors/>

- [18] Ajayvikram Chauhan; Gaurav Savner; Prajwal Venkatesh; Vishwanath Patil; Wencen Wu, A Blockchain-based Tracking System, doi: 10.1109/SOSE49046.2020.00020
- [19] Dinesh Ghorpade, Roshan Devkar, Vijay Sadgir, Sagar Ghodsare, Shibnath Nandi, Dnyaneshwar Mandlik, Tracking System Using Blockchain doi:10.1109/ICDCSW.2016.28.
- [20] Rishita Gawade, Aditi Kale, Snehal Mane and Ashutosh Koli, Government Fund Allocation and Tracking System using Blockchain International Journal of Multidisciplinary Innovative Research ISSN: 2583-0228
- [21] Daijun Zhang; Wei Chen, Insurance Funds Involved in the Affordable Housing Construction Investment for Risk 10.1109/BIFE.2011.72.
- [22] Taehwan Kim, Heechun Park, and Taewhan Kim, Senior Member, IEEE Allocation of Always On State Retention Storage for Power Gated Circuits—Steady-State-Driven Approach, doi: 10.1109/IC3INA.2016.7863031
- [23] Lin Li; Xuezhi Xia; Wei Chen, A Framework for Track Consistency Processing Based on Blockchain,
- [24] Meghna Vadher, Shivani Pandey, Darshana Sawant, Hezal Lopes, State government fund allocation and transaction system using blockchain technology, doi: www.jetir.org (ISSN 2349-5162)

A. SOURCE CODE

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.9;

contract funds {
    struct Campaign {
        address owner;
        string title;
        string
        description;
        uint256 target;
        uint256 deadline;
        uint256
        amountCollected; string
        image;
        address[] donators;
        uint256[] donations;
    }

    mapping(uint256 => funds) public campaigns;

    uint256 public numberOfCampaigns = 0;

    function createCampaign(address _owner, string memory _title,
string memory
_description, uint256 _target, uint256 _deadline, string memory _image) public
returns (uint256) {
        Campaign storage campaign = campaigns[numberOfCampaigns];

        require(campaign.deadline < block.timestamp, "The deadline should be a
date in the future.");

        campaign.owner = _owner;
        campaign.title = _title;
        campaign.description = _description;
        campaign.target = _target;
```

```
campaign.deadline = _deadline;  
campaign.amountCollected = 0;
```

```

campaign.image = _image;

numberOfrequests++;

return numberOfCampaigns - 1;
}

```

```

function donateToCampaign(uint256 _id) public payable {
    uint256 amount = msg.value;

    Campaign storage campaign = campaigns[_id];
    campaign.donators.push(msg.sender);
    campaign.donations.push(amount);
    (bool sent,) = payable(campaign.owner).call{value: amount}("");

    if(sent) {

        campaign.amountCollected = campaign.amountCollected + amount;
    }
}

```

```

function getDonators(uint256 _id) view public returns (address[] memory,
uint256[] memory) {
    return (campaigns[_id].donators, campaigns[_id].donations);
}

```

```

function getCampaigns() public view returns (Campaign[] memory) {
    Campaign[] memory allCampaigns =
    new
    Campaign[](numberOfCampaigns);

```

```
for(uint i = 0; i < numberOfCampaigns; i++) {
```

```

        Campaign storage item = campaigns[i];

        allCampaigns[i] = item;
    }

    return allCampaigns;
}
}

```

```

import React, { useState } from 'react'
import { Link, useNavigate } from 'react-router-dom';

import { useStateContext } from '../context';
import { CustomButton } from './.';
import { logo, menu, search, thirdweb } from '../assets';
import { navlinks } from '../constants';

const Navbar = () => {
    const navigate = useNavigate();
    const [isActive, setIsActive] = useState('dashboard');
    const [toggleDrawer, setToggleDrawer] =
    useState(false); const { connect, address } =
    useStateContext();

    return (
        <div className="flex md:flex-row flex-col-reverse justify-between
        mb-[35px] gap-6">
            <div className="lg:flex-1 flex flex-row max-w-[458px] py-2 pl-4 pr-2 h-
            [52px]
            bg-[#1c1c24] rounded-[100px]">

```

```
<input type="text" placeholder="Search for campaigns" className="flex w
full font-epilogue font-normal text-[14px] placeholder:text-[#4b5264] text-white
bg transparent outline-none" />
```

```
<div className="w-[72px] h-full rounded-[20px] bg-[#4acd8d] flex justify
center items-center cursor-pointer">
```

```
<img src={search} alt="search" className="w-[15px] h-[15px] object
contain"/>
```

```
</div>
```

```
</div>
```

```
<div className="sm:flex hidden flex-row justify-end gap-4">
```

```
<CustomButton
```

```
  btnType="button"
```

```
  title={address ? 'Create a campaign' : 'Connect'}
```

```
  styles={address ? 'bg-[#1dc071]' :
```

```
'bg-[#8c6dfd]'} handleClick={() => {
```

```
    if(address)
```

```
      navigate('create-campaign')    else
```

```
      connect()
```

```
  }}
```

```
/>
```

```
<Link to="/profile">
```

```
<div className="w-[52px] h-[52px] rounded-full bg-[#2c2f32] flex justify
center items-center cursor-pointer">
```

```
<img src={thirdweb} alt="user" className="w-[60%] h-[60%] object
contain" />
```

```
</div>
```

```
</Link>
```

```
</div>
```

```

{ /* Small screen navigation */}
  <div className="sm:hidden flex justify-between items-center relative">
    <div className="w-[40px] h-[40px] rounded-[10px] bg-[#2c2f32] flex justify
center items-center cursor-pointer">
      <img src={logo} alt="user" className="w-[60%] h-[60%] object-contain"
    />
  </div>

```

```

<img
  src={menu}
  alt="menu"
  className="w-[34px] h-[34px] object-contain cursor-pointer"
  onClick={() => setToggleDrawer((prev) => !prev)}
/>

```

```

<img
  src=
  {
    d
    a
    s
    h
  }
  alt=
  "
  r
  e

```

q
u
e
s
t
"

```
className="w-[34px] h-[34px]  
object-contain cursor-pointer"  
onClick={() =>  
  setToggleDrawer((prev) => !prev)}  
/>
```

```
<div className={`absolute top-[60px] right-0 left-0 bg-[#1c1c24] z-10  
shadow-secondary py-4 ${!toggleDrawer ? '-translate-y-[100vh]' : 'translate-y-0'}  
transition-all duration-700`}>  
  <ul className="mb-4">  
    {navlinks.map((link) => (
```

```

<li
  key={link.name}
  className={`flex p-4 ${isActive === link.name && 'bg-[#3a3a43]'}`}
  onClick={() => {
    setIsActive(link.name);
    setToggleDrawer(false);
    navigate(link.link);
  }}
>

  <img
    src={link.imageUrl}
    alt={link.name}
    className={`w-[24px] h-[24px] object-contain ${isActive ===
link.name ? 'grayscale-0' : 'grayscale'}`}
  />

  <p className={`ml-[20px] font-epilogue font-semibold text-[14px]
${isActive === link.name ? 'text-[#1dc071]' : 'text-[#808191]'}`}>
    {link.name}
  </p>
</li>
)}}
</ul>

```

```

<div className="sm:flex hidden flex-row justify-end gap-4">
  <CustomButton
    btnType="button"
    title={address ? 'Create a campaign' : 'Connect'}
    styles={address ? 'bg-[#1dc071]' :
'bg-[#8c6dfd]'} handleClick={() => {
      if(address)
        navigate('create-campaign')
      else
        connect()
    }}
  >

```

/>

```

<Link to="/profile">
  <div className="w-[52px] h-[52px] rounded-full bg-[#2c2f32] flex justify
center items-center cursor-pointer">
    <img src={thirdweb} alt="user" className="w-[60%] h-[60%] object
contain" />
  </div>
</Link>
</div>

```

```

<div className="flex mx-4">

<img src=
{
  link
  .
  img
  Url
}
alt=
{

```

```

l                                a
i                                m
n                                e
k                                }
.                                className={`w-[24px] h-[24px]
n                                object-contain ${isActive ===

link.name ? 'grayscale-0' : 'grayscale'}`}
    />
    <p className={`ml-[20px] font-epilogue font-semibold text-[14px]
${isActive === link.name ? 'text-[#1dc071]' : 'text-[#808191]'}>
      {link.name}
    </p>
  </li>
)}}
</ul>
  <CustomButton
    btnType="button"
    title={address ? 'Create a campaign' : 'Connect'}
    styles={address ? 'bg-[#1dc071]' :
      'bg-[#8c6dfd]'} handleClick={() => {
      if(address)
        navigate('create-campaign')      else
        connect();
    }}
  />
</div>

```

```
        </div>
      </div>
    </div>
  )
}

export default Navbar
```

B. SCREENSHOTS

```
PS C:\Users\Ranvi\Downloads\project_crowdfunding-master\project_crowdfunding-master\web3> npm run
Scripts available in hardhat-javascript-starter via `npm run-script`:
  build
    npx thirdweb@latest detect
  deploy
    npx thirdweb@latest deploy
  release
    npx thirdweb@latest release

PS C:\Users\Ranvi\Downloads\project_crowdfunding-master\project_crowdfunding-master\web3> npx thirdweb@latest deploy
Need to install the following packages:
thirdweb@5.3.0
Ok to proceed? (y) y
[#####.....] - idealTree:solc: 5.1.1 placeDep ROOT semver@7.6.0 OK for: @eth-optimism/sdk@3.2.3 want: ^7.6.0
```

Fig. B.1. Deployment Execution

[illegible]

Fig. B.2. Successful Deployment

Ethereum Mainnet





Welcome back!

The decentralized web awaits

Password

Unlock

[Forgot password?](#)

Need help? Contact [MetaMask support](#)

Fig. B.3 Metamask Login

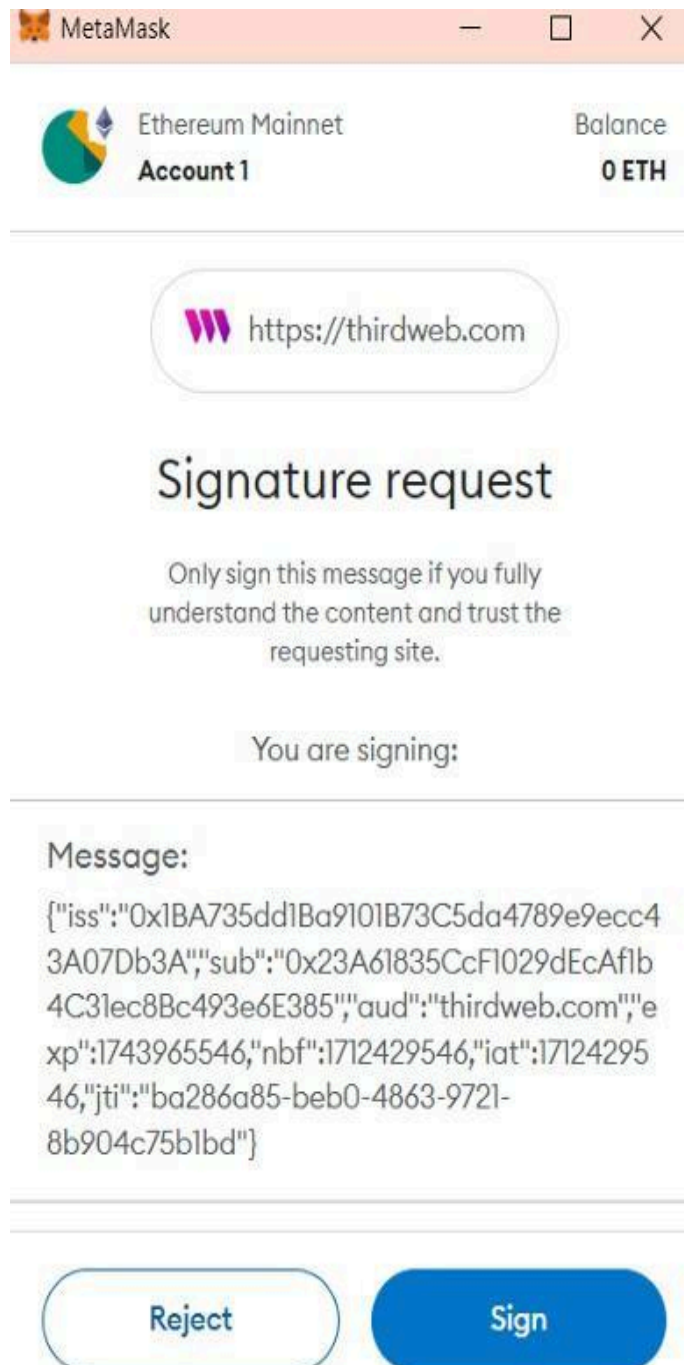


Fig. B.4 Metamask Connection Request

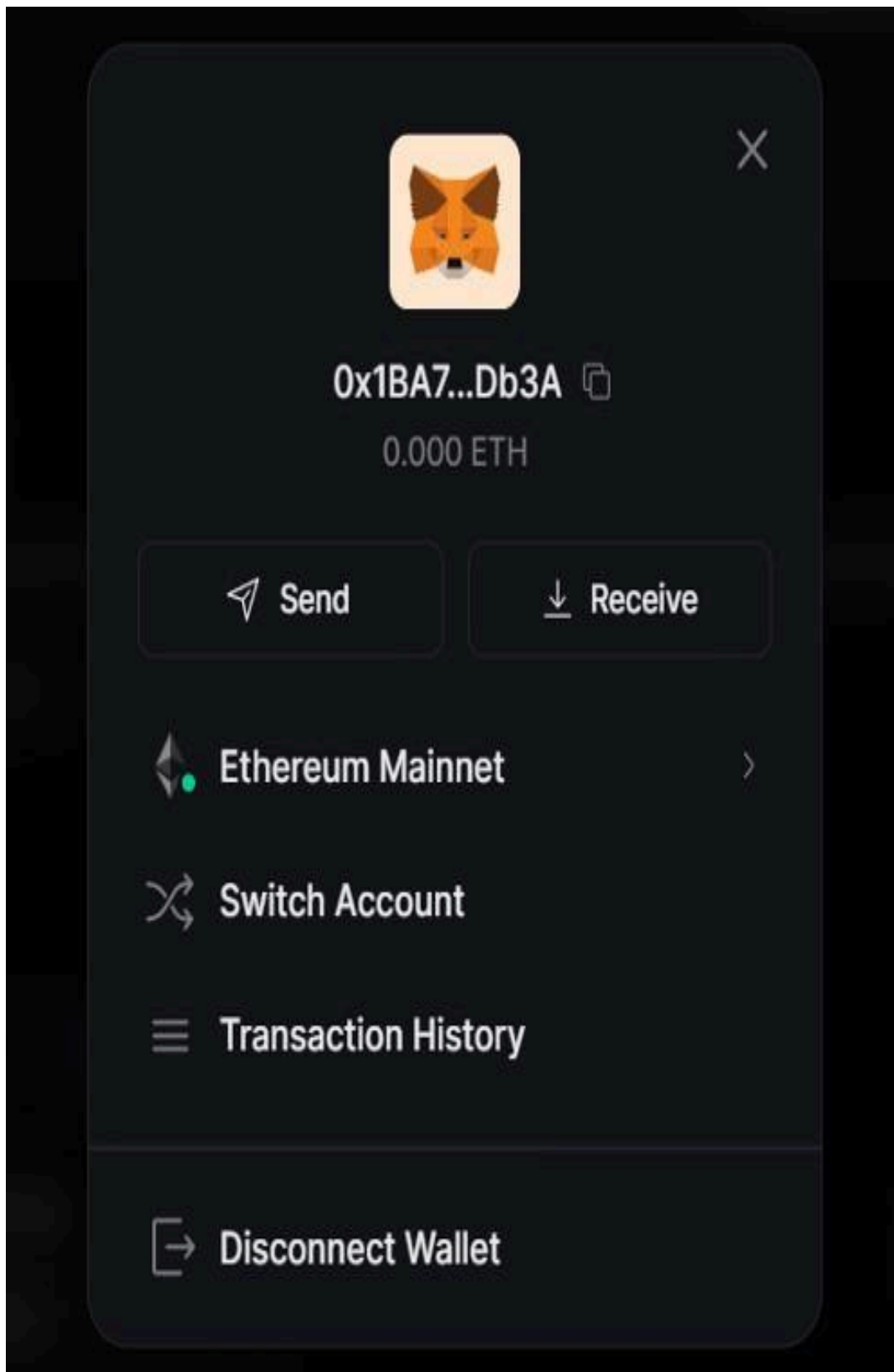


Fig B.5 Metamask connection in thirdweb

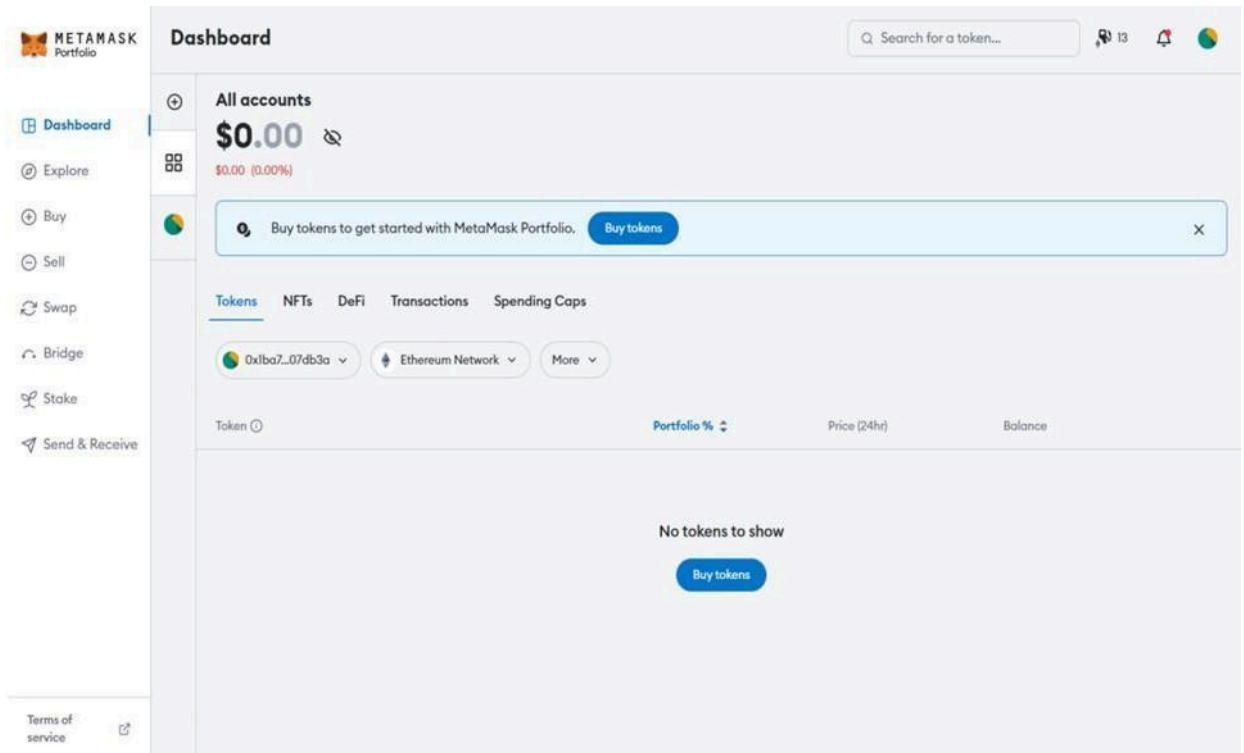


Fig. B.6 Metamask Dashboard