TRANSPARENT BLOCKS

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

By

Mudunuri Sai Siddhartha Varma(Reg.No-39110646)
Tamalala Venkata Siva Bhanu Teja(Reg. No-39111016)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119

APRIL 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Mudunuri Sai Siddhartha Varma (Reg.No: 39110646) and Tamalala Venkata Siva Bhanu Teja (Reg.No: 39111016)** who carried out the project Phase -2 entitled **"TRANSPARENT BLOCKS"** under my supervision from Jan 2023 to April 2023.

Internal Guide
Ms. Nancy Kirupanithi.D B.E.,M.Tech.,

Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva-voce Examination held on 20.04.2023

Internal Examiner **External Examiner**

DECLARATION

I,**Mudunuri Sai Siddhartha Varma (Reg.No : 39110646),Tamalala Venkata Siva Bhanu Teja (Reg.No : 39111016)** hereby declare that the project report entitled **Transparent Blocks** was done by me under the guidance of **Ms.NancyKirupanithi.D B.E.,M.TECH** andis submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in **Computer Science and Engineering**.

Date : 20.04.2023

Sai sidhartha varma

PLACE :Chennai **Signature of the Student**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **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. PhD, Dean**, School of Computing, **Dr S**. **Vigneshwari, M.E. PhD** and **Dr L. Lakshmanan, M.E., PhD**, Heads 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.NancyKirupanithi.D B.E.,M.TECH .** for her valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of my 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

The Government of India receives a lot of money in the form of Income tax, GST.etc. and as a common man, there is no way for us to know the amount of money they receive through taxes and also where the money is being spent. And also Corruption in India is an issue which affects the economy of central, state and local government agencies in many ways. The causes of corruption are mainly due to a lack of transparency and traceability. The proposed project using blockchain can bring transparency and traceability and helps people to see the amount of money that is being received and transferred by the government, hence reducing corruption.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
UI	User Interface
RAM	Random Access Memory
OS	Operating System
MACOS	Macintosh Operating System
GB	GigaByte
ERC	Ethereum request for comment
GST	Goods and Service Tax
POS	Proof-of-Stake
POW	Proof-of-Work
IDE	Integrated Development Environment
EVM	Ethereum Virtual Machine
WWW	World Wide Web
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
BSD	Berkeley Software Distribution
HTTP	HyperText Transfer Protocol

CHAPTER 1
INTRODUCTION

1.1 GENERAL DESCRIPTION

The costs to society of public-sector corruption and weak accountability are staggering. In many parts of the world, public-sector corruption is the single-largest challenge, stifling social, economic and environmental development. Often, corruption centres around a lack of transparency, inadequate record-keeping and low public accountability. Blockchain and distributed ledger technologies, when applied thoughtfully to certain corruption-prone government processes, can potentially increase transparency and accountability in these systems, reducing the risk or prevalence of corrupt activity.

1.1.1 PRODUCT DESCRIPTION:

Transparent Blocks is a project where a token is created using smart contracts deployed on the Ganache network and a block explorer which allows users or peers to see all the transactions that are made on the network to increase transparency.

1.1.2 PROBLEM STATEMENT:

The Government of India receives a lot of money in the form of Income tax, GST.etc. and as a common man, there is no way for us to know how much money they receive through taxes and also where the money is being used. And also Corruption in India is an issue which affects the economy of central, state and local government agencies in many ways. The causes of corruption are mainly due to a lack of transparency and traceability. The proposed project using blockchain can bring transparency and traceability and helps people to see the amount of money that is being received and transferred by the government, hence reducing corruption.

1.2 SYSTEM OBJECTIVES

- Improvement in control and performance
    - The system is developed to cope with the current issues and problems and is bug-free.
    - Blockchain is one of the most powerful and promising technologies nowadays in the IT industry, One of Blockchain's main features is the presence of a Consensus Algorithm, which is responsible for maintaining the security and integrity of the entire blockchain network where all the nodes participating in the network reach a certain agreement.

- Security
    - The blockchain is immutable i.e once a block is verified and added to the chain , it is impossible to modify the block. Also it is impossible to hack the blockchain
    - Blockchain technology achieves decentralized security and trust in several ways. To begin with, new blocks are always stored linearly and chronologically. That is, they are always added to the "end" of the blockchain.
    - After a block has been added to the end of the blockchain, it is extremely difficult to go back and alter the contents of the block unless a majority of the network has reached a consensus to do so.
    - That's b That's because each block contains its own hash, along with the hash of the block before it, as well as the previously mentioned timestamp. because each block contains its own hash, along with the hash of the block before it, as well as the previously mentioned timestamp.
    - Let's say that a hacker, who also runs a node on a blockchain network, wants to alter a blockchain and steal cryptocurrency from everyone else. If they were to alter their own single copy, it would no longer align with everyone else's copy. When everyone else cross-references their copies against each other, they would see this one copy stand out,

    and that hacker’s version of the chain would be cast away as illegitimate.
    - Succeeding with such a hack would require that the hacker simultaneously control and alter 51% or more of the copies of the blockchain so that their new copy becomes the majority copy and, thus, the agreed-upon chain. Such an attack would also require an immense amount of money and resources, as they would need to redo all of the blocks because they would now have different timestamps and hash codes.
    - Due to the size of many cryptocurrency networks and how fast they are growing, the cost to pull off such a feat probably would be insurmountable. This would be not only extremely expensive but also likely fruitless. Doing such a thing would not go unnoticed, as network members would see such drastic alterations to the blockchain. The network members would then hard fork off to a new version of the chain that has not been affected. This would cause the attacked version of the token to plummet in value, making the attack ultimately pointless, as the bad actor has control of a worthless asset. The same would occur if the bad actor were to attack the new fork of Bitcoin. It is built this way so that taking part in the network is far more economically incentivized than attacking it.
- Transparency and Traceability
    - The transactions that happen on the blockchain are visible to everyone, thus increasing transparency and traceability.
    - Because of the decentralized nature of Bitcoin’s blockchain, all transactions can be transparently viewed by either having a personal node or using Block Explores that allow anyone to see transactions occurring live. Each node has its own copy of the chain that gets updated as fresh blocks are confirmed and added. This means that if you wanted to, you could track Bitcoin wherever it goes.
    - For example, exchanges have been hacked in the past, where those who kept Bitcoin on the exchange lost everything. While the hacker

may be entirely anonymous, the Bitcoins that they extracted are easily traceable. If the Bitcoins stolen in some of these hacks were to be moved or spent somewhere, it would be known.

- Save time
    - The smart contract is deployed over the Ethereum chain, which has relatively less block time i.e verification and completion transactions take less time.

    - Blockchain is a type of shared database that differs from a typical database in the way that it stores information; blockchains store data in blocks that are then linked together via cryptography. As new data comes in, it is entered into a fresh block. Once the block is filled with data, it is chained onto the previous block, which makes the data chained together in chronological order.
    - Different types of information can be stored on a blockchain, but the most common use so far has been as a ledger for transactions. In Bitcoin's case, blockchain is used in a decentralized way so that no single person or group has control-rather, all users collectively retain control.
    - Decentralized blockchains are immutable, which means that the data entered is irreversible. For Bitcoin, this means that transactions are permanently recorded and viewable to anyone.

- Smart Contracts

    - A smart contract is a computer code that can be built into the blockchain to facilitate, verify, or negotiate a contract agreement. Smart contracts operate under a set of conditions to which users agree. When those conditions are met, the terms of the agreement are automatically carried out.
    - Say, for example, that a potential tenant would like to lease an apartment using a smart contract. The landlord agrees to give the tenant the door code to the apartment as soon as the tenant pays the security deposit. Both the tenant and the landlord would send their respective portions of the deal to the smart contract, which would hold onto and automatically exchange the door code for the security deposit on the date when the lease begins. If the landlord doesn’t supply the door code by the lease date, then the smart contract refunds the security deposit. This would eliminate the fees and processes typically associated with the use of a notary, a third-party mediator, or attorneys.

CHAPTER-2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

Currently, in India, Corruption is an issue which affects the economy of central, state and local government agencies in many ways. Corruption is blamed for stunting the Economy of India. The causes of corruption in India include excessive regulations, complicated tax and licensing systems, numerous government departments, a monopoly of government-controlled institutions on certain goods and services delivery, and the lack of transparent laws and processes. Corruption activities can only be detected through Self-reporting, Citizen reporting or Journalism and media reporting.

Also, there is no way for a common man to see the amount of money the government of India receives through taxes and also where the money is being used except through the physical records which are maintained and are not accessible to everyone. The proposed project using blockchain can bring transparency and traceability and helps people to see the amount of money that is being received and transferred by the government, hence reducing corruption.

2.2 PROPOSED SYSTEM

In this system we are going to create and deploy a smart contract over the ethereum chain such that the smart contract is able to process, validate and complete the transactions efficiently using wallet to store and send our tokens based on certain conditions, the system allows only verified users to send and receive tokens, along with a user interface where the users can verify their wallet address, and view all the transactions happening over the blockchain. Thus increasing transparency and traceability.

Advantages:

- Less Block time
- Decentralisation and Security
- Less gas fee
- Proof of Stake (PoS) consensus mechanism better than Pow
- Transparency and traceability

CHAPTER-3
AIM AND SCOPE OF THE PRESENT INVESTIGATION

This chapter gives an overview of the aim, objectives, background and operation environment of the system.

3.1 PROJECT AIMS AND OBJECTIVES

The project aims and objectives that will be achieved after the completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Create a smart contract using solidity that can be deployed over the ethereum network.
- Create a token that can be transacted using wallets such as Wallet.
- Create a block explorer that reads the transactions and displays them.
- Create a user interface which allows users to verify their wallet address and also to view all the transactions on the blockchain.

3.2 SYSTEM REQUIREMENTS

3.2.1 NON-FUNCTIONAL REQUIREMENTS

Product Requirements

3.2.1.1 EFFICIENCY REQUIREMENT

The system should efficiently make and display the transactions.

3.2.1.2 RELIABILITY REQUIREMENT

The system should verify the transactions accurately.

3.2.1.3 USABILITY REQUIREMENT

The system should perform various tasks easily and in an effective way.

3.2.1.4 IMPLEMENTATION REQUIREMENT

In the implementation, the application uses tools such as solidity, Wallet, goerli test net, python, block explorer etc.

3.2.1.5 DELIVERY REQUIREMENT

The whole system is expected to be delivered in 2 months with a weekly evaluation by the project guide.

3.2.2 FUNCTIONAL REQUIREMENTS

3.2.2.1 VERIFY USER

Description of feature:

This feature is used by the users to verify their wallet addresses.

3.2.2.2 VERIFY AND COMPLETE MULTIPLE TRANSACTIONS

Description of feature:

The smart contract will be able to process, verify and complete transactions efficiently.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system

3.3.1 SOFTWARE REQUIREMENTS

- Operating system- Windows, Linux or MAC operating system is used as it is stable and supports more features and is more user-friendly.
- Development tools such as solidity and goerlitestnet and Wallet.

3.3.2 HARDWARE REQUIREMENTS

- Any processor can run the program and also is reliable and stable
- A minimum of 2 GB of RAM is recommended.

PROCESSOR	Any processor
OPERATING SYSTEM	Windows 8, Windows 10, MAC OS, Linux, Chrome OS
MEMORY	2 GB RAM or higher,
HARD DISK SPACE	Minimum 4 GB or higher
GRAPHICS CARD	Not Mandatory
INTERNET	Must have a good internet connection

CHAPTER 4

EXPERIMENTAL MATERIALS AND METHODS AND ALGORITHMS USED

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

- A Smart contract written in solidity is deployed over the ethereum network which creates a token that can be transacted from one address to another. The advantages of creating a token using smart contracts are
    - Less Block time
    - Less gas fee
    - Faster Transactions
    - Proof of Stake (POS) consensus mechanism

- The goerliethereumtestnet is used to deploy the smart contract as it provides a safe environment to run, test and debug the smart contracts. The advantages are:
    - A testnet node needs to sync and store much fewer data (depending on the network)
    - Testnets are faster with transactions, and are almost instantaneous.
    - Deploying contracts or making transactions requires test ether, which has no value and can be acquired for free from several “faucets”.

- Wallet is used to store the tokens that are created through smart contracts as it is secure and can be used to transfer the tokens using digital signatures. The advantages are
    - Easy to use simple user interface
    - Secure
    - Easy to Backup and restore
    - Supports multiple tokens
- Python, flask, Html, and CSS are used to create a user interface which facilitates user interaction. Html and CSS are used for the front end and Python is used for the back end which runs the Html, and CSS templates using Flask and also manage the data required for the application.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

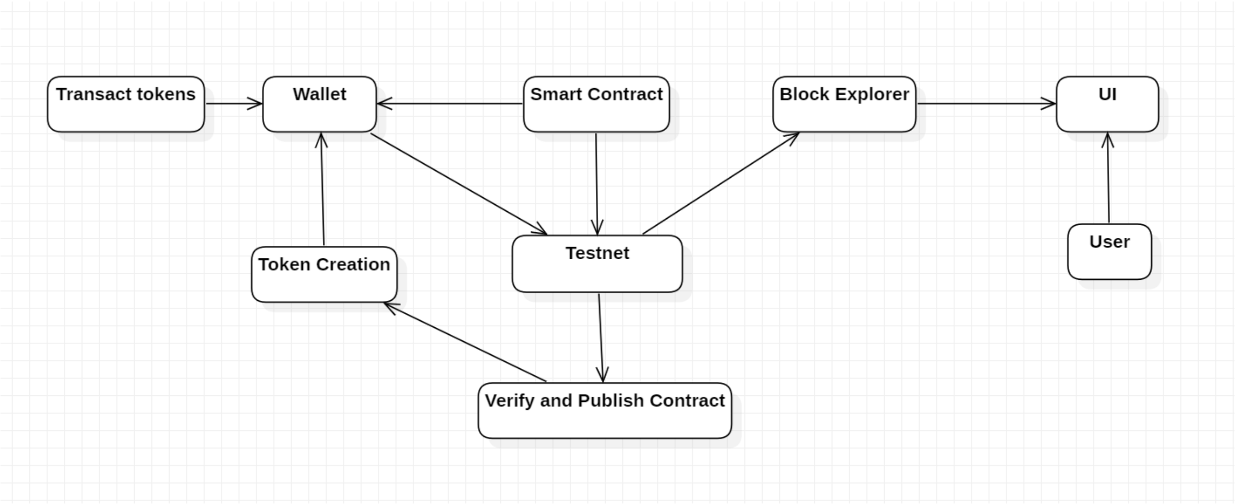


Fig 4.2: **System Architecture for Transparent Blocks**

4.3 SOFTWARE TOOLS USED

4.3.1 SOLIDITY

Solidity is an object-oriented programming language for implementing smart contracts on various blockchain platforms, most notably, Ethereum. It was developed by Christian Reitwiessner, Alex Beregszaszi, and several former Ethereum core contributors. Programs in Solidity run on Ethereum Virtual Machine.

Solidity is a statically typed programming language designed for developing smart contracts that run on the Ethereum Virtual Machine (EVM).
Smart contracts are high-level program codes that are compiled to EVM byte code and deployed to the Ethereum blockchain for further execution. It allows us to perform credible transactions without any interference of the third party, these transactions are trackable and irreversible. Languages used to write smart contracts are Solidity (a language library with similarities to C and JavaScript), Serpent (similar to Python, but deprecated), LLL (a low-level Lisp-like language), and Mutan (Go-based, but deprecated)

4.3.2 REMIX

Remix IDE is an open-source web and desktop application. It fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. Remix is used for the entire journey of contract development with Solidity language as well as a playground for learning and teaching Ethereum.

Remix IDE is part of the Remix Project which also includes the Remix Plugin Engine and Remix Libraries: low-level tools for wider use.

Remix IDE is used for the entire journey of smart contract development by users at every knowledge level. It requires no setup, fosters a fast development cycle, and has a rich set of plugins with intuitive GUIs. The IDE comes in two flavors (web app or desktop app) and as a VSCode extension.

Supported browsers: Firefox, Chrome, Brave. We do not support use of Remix on tablets or mobile devices.
This set of documents covers instructions on how to use Remix, LearnEth located inside of Remix IDE

4.3.3 Wallet

Wallet is a software cryptocurrency 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 decentralised applications.

Wallet allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralised applications through a compatible web browser or the mobile app's built-in browser.

4.3.4 TESTNET

In blockchain technology, a testnet is an instance of a blockchain-powered by the same or a newer version of the underlying software, to be used for testing and experimentation without risk to real funds or the main chain. Testnet coins are separate and distinct from the official (mainnet) coins, don't have value, and can be obtained freely from faucets.

Testnets allow for the development of blockchain applications without the risk of losing funds.

4.3.5 PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasises code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects [magic methods]. Many other paradigms are supported via extensions, including design by contract and logic programming.

4.3.6 FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Flask is often referred to as a microframework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application.

4.3.7 HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.

4.3.8 *CSS*

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web(WWW), alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. CSS file, which reduces complexity and repetition in the structural content; and enables the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

4.3.9 JINJA2

Jinja is a web template engine for the Python programming language. It was created by Armin Ronacher and is licensed under a BSD Licence. Jinja is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated in a sandbox. It is a text-based template language and thus can be used to generate any markup as well as source code.

The Jinja template engine allows the customization of tags, filters, tests, and globals. Also, unlike the Django template engine, Jinja allows the template designer to call functions with arguments on objects. Jinja is Flask's default template engine and it is also used by Ansible, Trac, and Salt.

4.4.1 Ganache

The scale of developments in the world of web3 has been multiplying by huge margins. With the number of users in the domain of web3 continuing to increase radically, the adoption of dApps and web3 technologies has also witnessed an upward trend. On the other hand, the process of web3 is unreasonably complicated for various reasons. Therefore, tools such as the Ganache blockchain have made a mark in the domain of smart contract development on the grounds of productivity improvements.

CHAPTER-5

RESULTS AND DISCUSSION ,PERFORMANCE ANALYSIS

5.1 RESULTS

The benefits of blockchain in Transactions and Transparency include:

1. **Security**: Blockchain technology provides a secure way to store Transaction history ,reducing the risk of fraud or unauthorized access.
2. **Transparency**: The decentralized nature of blockchain allows for a transparent verification of Wallet address , which can be accessed and verified by authorized parties.
3. **Efficiency**: Blockchain technology can streamline the verification process, reducing the time and resources required to verify Wallet address.
4. **Scalability**: Blockchain technology can be easily scaled to accommodate a large number of Transactions , making it an ideal solution for others to view .

5.2 DISCUSSION

However, there are also some potential challenges to implementing blockchain technology in Transaction and transparency including:

1. **Adoption**: Wide adoption of blockchain technology among Transactions may take some time, which could slow down the implementation of this solution.
2. **Technical challenges**: Implementation of blockchain requires significant technical expertise and investment, which may not be feasible for all Users.
3. **Privacy concerns**: While blockchain technology can provide secure storage of Transactions it may also raise concerns about privacy and data protection.Despite these challenges, the use of blockchain technology in wallet to wallet transactions has the potential to greatly benefit to both Sender and Recevier and institutions by providing a more secure, efficient, and scalable method of Transactions. As such, it is an area that is likely to continue to be explored and developed in the coming years.

5.3 SYSTEM TESTING

The system testing process aimed to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made. Based on these observations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing.

5.3.1 UNIT TESTING

- Unit testing is undertaken when a module has been created and successfully reviewed in order to test a single module we need to provide a complete environment i.e. beside the module we would require.

- The procedures belonging to other modules that the module under test calls
- A procedure to call the functions of the module under test with appropriate Parameters
- Non-local data structures that the module accesses

Unit testing was done on each and every module that is described under the module description.

Test For the USER INTERFACE module

- Testing User Interface such that it allows users to verify their account, search for an account, and view the transactions happening.

Test For the SMART CONTRACT module

- The Smart contract is deployed and the token is created successfully.
-

5.4 INTEGRATION TESTING

- In this type of testing, we test various integration of the project module by providing the input. The primary objective is to test the module interfaces to ensure that no errors occur when one module invokes the other module.

SUMMARY AND CONCLUSION

The Smart Contract is created using solidity to create a token such that the token can only be sent to a verified user is compiled successfully and is deployed on the ganache Ethereum testnet and wallet is used to store the and transfer the token. And a user interface is created on the web using python, flask, Html, Css which allows users to verify their account so that they can start receiving and transferring the token, search the wallet addresses of the users based on name or Id, and view all the transactions on the blockchain.

REFERENCES:-

1.Saleh, O.S.; Ghazali, O.; Rana, M.E. Blockchain-based framework for educational certificates verification. *J. Crit. Rev.* **2020**, *7*, 79–84. [GoogleScholar] [CrossRef]

2.Muzammil, M. Corrupt schools, corrupt universities: What can be done? *Comp. A J. Comp. Int. Educ.* **2010**, *40*, 385–387. [Google Scholar] [CrossRef]

2.Creating Pathways to Careers in IT. Available online:https://services.google.com/fh/files/misc/it_cert_impactreport_booklet_rgb_digital_version.pdf.(accessed on 16 March 2021).

4.Lyons, T.; Courcelles, L.; Timsit, K. *Blockchain and the GDPR*; The European Union Blockchain Observatory and Forum, European Commission: Brussels, Belgium, 2018. [Google Scholar]

5.Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online:https://www.bitcoin.org/bitcoin.pdf.(accessed on 17 January 2021).

6.Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [Google Scholar] [CrossRef]

7.Mirabelli, G.; Solina, V. Blockchain and agricultural supply chains traceability: Research trends and future challenges. *Procedia Manuf.* **2020**, *42*, 414–421. [Google Scholar] [CrossRef]

8.Khezr, S.; Moniruzzaman, M.; Yassine, A.; Benlamri, R. Blockchain Technology in Healthcare: A Comprehensive Review and Directions for Future Research. *Appl. Sci.* **2019**, *9*, 1736. [Google Scholar] [CrossRef][GreenVersion]

9.Jiang, S.; Cao, J.; Wu, H.; Yang, Y.; Ma, M.; He, J. BlocHIE: A BLOCkchain-Based Platform for Healthcare Information Exchange. In Proceedings of the 2018 IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, Sicily, Italy, 18–20 June 2018; Institute of

Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 49–56. [Google Scholar] [CrossRef]

10.Pournader, M.; Shi, Y.; Seuring, S.; Koh, S.L. Blockchain applications in supply chains, transport and logistics: A systematic review of the literature. *Int. J. Prod. Res.* **2019**, *58*, 2063–2081. [Google Scholar] [CrossRef]

11.Conoscenti, M.; Vetro, A.; de Martin, J.C. Blockchain for the Internet of Things: A systematic literature review. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 1–6. [GoogleScholar] [CrossRef][Green Version]

12.Jiang, S.; Cao, J.; Wu, H.; Yang, Y. Fairness-based Packing of Industrial IoT Data in Permissioned Blockchains. *IEEE Trans. Ind. Inform.* **2020**, 1. [GoogleScholar] [CrossRef]

13.Konstantinidis, I.; Siaminos, G.; Timplalexis, C.; Zervas, P.; Peristeras, V.; Decker, S. *Blockchain for Business Applications: A Systematic Literature Review*; Abramowicz, W., Paschke, A., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 320, pp. 384–399. [Google Scholar]

14.Butijn, B.-J.; Tamburri, D.A.; Heuvel, W.-J.V.D. Blockchains. *ACM Comput. Surv.* **2020**, *53*, 1–37. [Google Scholar] [CrossRef]

15.Casino, F.; Dasaklis, T.K.; Patsakis, C.F. A systematic literature review of blockchain-based applications: Current status, classification and open issues. In *Telematics and Informatics*; Elsevier Ltd: Amsterdam, The Netherlands, 2019; Volume 36, pp. 55–81. [Google Scholar] [CrossRef]

APPENDIX

A. SOURCE CODE

Python

main.py

```
import variables
import json
from flask import *
from chain import Chain


def block_div(block_no, from_name, from_address, to_name, to_address, value):
    block = f"""<div class="block b{block_no}">\n<div class="block-title">Block
#{block_no}</div>\n<div class="block-details">\n<div>From :
{from_name}({from_address})</div>\n<div>To:
{to_name}({to_address})</div>\n<div>Value :{value}</div>\n</div>\n</div>"""
    return block


app = Flask(__name__)
block_chain = Chain()


@app.route("/")
def explorer():
    with open("./data/blocks.json") as file:
blocks_data = json.load(file)
blocks_data = {i: blocks_data[str(i)] for i in sorted([int(i) for i in
list(blocks_data.keys())])}
    with open("./data/accounts.json") as file:
        accounts = json.load(file)
all_blocks = ""
    for i in list(blocks_data.keys()):
```

```
from_address = blocks_data[i]["from"]
from_name = accounts.get(from_address, "")
to_address = blocks_data[i]["to"]
to_name = accounts.get(to_address, "")
        value = blocks_data[i]["value"]
all_blocks += block_div(i, from_name, from_address, to_name, to_address, value)
all_blocks = "{% extends 'explorer.html' %}\n{% block content %}\n" + all_blocks +
"\n{% endblock %}"
    with open('./templates/blocks.html', 'w+') as file:
file.write(all_blocks)
    return render_template("blocks.html")


@app.route("/wallet")
def wallet():
    if variables.logged_in:
        return render_template("wallet.html",
name=variables.current_account_name,
                               address=variables.current_account_address,

balance=block_chain.get_balance(variables.current_account_address),
symbol="INR")
    return render_template("walletLogin.html")


@app.route("/send")
def send():
    to = request.args.get('to')
    value = request.args.get('amount')
block_chain.transfer(variables.current_account_address, to, value)
    with open("./data/blocks.json") as file:
block_no, block_from = block_chain.get_latest_block()
        blocks = json.load(file)
        blocks[str(block_no)] = {"from": block_from, 'to': to, "value": value}
    with open("./data/blocks.json", 'w+') as file:
json.dump(blocks, file)
    return redirect("/wallet")
```

```
@app.route("/login-page")
def login_page():
variables.logged_in = False
    return render_template("walletLogin.html")


@app.route("/login")
def login():
    username = request.args.get("username")
    password = request.args.get("password")
    with open("./data/credentials.json") as file:
        creds = json.load(file)
        if password and creds.get(username).get("password") == password:
variables.logged_in = True
variables.current_account_name = username
variables.current_account_address = creds.get(username).get("address")
            return redirect("/wallet")
    return redirect("/login-page")


@app.route("/create-account")
def create_account():
    return render_template("createWallet.html")


@app.route("/signup")
def sign_up():
    username = request.args.get("username")
    password = request.args.get("password")
retyped_password = request.args.get('retype-password')
    if (password != retyped_password) and password:
        return redirect('/create-account')
assigned_address = block_chain.assign_account(username)
    with open("./data/credentials.json") as file:
        creds = json.load(file)
        creds[username] = {"password": password, "address": assigned_address}
    with open("./data/credentials.json", 'w+') as file:
json.dump(creds, file)
    return redirect('/login-page')


app.run()
```

variables.py

```
abi = [
    {
        "anonymous": False,
        "inputs": [
            {
                "indexed": True,
                "internalType": "address",
                "name": "tokenOwner",
                "type": "address"
            },
            {
                "indexed": True,
                "internalType": "address",
                "name": "spender",
                "type": "address"
            },
            {
                "indexed": False,
                "internalType": "uint256",
                "name": "tokens",
                "type": "uint256"
            }
        ],
        "name": "Approval",
        "type": "event"
    },
{
    "anonymous": False,
    "inputs": [
        {
            "indexed": True,
            "internalType": "address",
            "name": "from",
            "type": "address"
        },
        {
            "indexed": True,
            "internalType": "address",
            "name": "to",
            "type": "address"
        },
        {
            "indexed": False,
            "internalType": "uint256",
            "name": "tokens",
            "type": "uint256"
        }
    ],
    "name": "Transfer",
    "type": "event"
},
{
    "constant": True,
    "inputs": [
        {
            "internalType": "address",
            "name": "tokenOwner",
            "type": "address"
        },
        {
            "internalType": "address",
            "name": "spender",
            "type": "address"
        }
    ],
    "name": "allowance",
    "outputs": [
        {
            "internalType": "uint256",
            "name": "remaining",
            "type": "uint256"
        }
    ],
    "payable": False,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": False,
    "inputs": [
        {
            "internalType": "address",
            "name": "spender",
            "type": "address"
        },
        {
            "internalType": "uint256",
            "name": "tokens",
            "type": "uint256"
        }
    ],
    "name": "approve",
    "outputs": [
        {
            "internalType": "bool",
            "name": "success",
            "type": "bool"
        }
    ],
    "payable": False,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": True,
    "inputs": [
        {
            "internalType": "address",
            "name": "tokenOwner",
            "type": "address"
        }
    ],
    "name": "balanceOf",
    "outputs": [
        {
            "internalType": "uint256",
            "name": "balance",
            "type": "uint256"
        }
    ],
    "payable": False,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": True,
    "inputs": [],
    "name": "totalSupply",
    "outputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "payable": False,
    "stateMutability": "view",
"type": "function"
},
{
"constant": False,
"inputs": [
    {
        "internalType": "address",
        "name": "to",
        "type": "address"
    },
    {
        "internalType": "uint256",
        "name": "tokens",
        "type": "uint256"
    }
],
"name": "transfer",
"outputs": [
    {
        "internalType": "bool",
        "name": "success",
        "type": "bool"
    }
],
"payable": False,
"stateMutability": "nonpayable",
"type": "function"
},
{
"constant": False,
"inputs": [
    {
        "internalType": "address",
        "name": "from",
        "type": "address"
    },
    {
            "internalType": "address",
            "name": "to",
            "type": "address"
        },
        {
            "internalType": "uint256",
            "name": "tokens",
            "type": "uint256"
        }
    ],
    "name": "transferFrom",
    "outputs": [
        {
            "internalType": "bool",
            "name": "success",
            "type": "bool"
        }
    ],
    "payable": False,
    "stateMutability": "nonpayable",
    "type": "function"
}
]

contract_address = "address"

logged_in = False

current_account_name = ""
current_account_address = "xxxxxx"
```

chain.py

```
import json
import random
from web3 import *
from variables import abi, contract_address


class Chain:

    def __init__(self):
self.chain = Web3(Web3.HTTPProvider("http://127.0.0.1:7545"))
self.contract = self.chain.eth.contract(address=contract_address, abi=abi)

    def get_balance(self, address):
        return self.contract.caller().balanceOf(address)

    def transfer(self, from_address, to_address, amount):
self.chain.eth.default_account = from_address
        return self.contract.functions.transfer(to=to_address,
tokens=int(amount)).transact()

    def get_latest_block(self):
latest_block_no = self.chain.eth.get_block_number()
block_data = self.chain.eth.get_block(latest_block_no)
        transaction = self.chain.eth.get_transaction(block_data['transactions'][0])
        return latest_block_no, transaction['from']

    def get_accounts(self):
        return set(self.chain.eth.accounts)

    def assign_account(self, name):
        accounts = self.get_accounts()
        with open("./data/accounts.json") as file:
            assigned = json.load(file)
assigned_accounts = set(assigned.keys())
free_accounts = accounts - assigned_accounts
to_assign_account = random.choice(list(free_accounts))
            assigned[to_assign_account] = name
        with open("./data/accounts.json", 'w+') as file:
json.dump(assigned, file)
        return to_assign_account


chain = Web3(Web3.HTTPProvider("http://127.0.0.1:7545"))
contract = chain.eth.contract(address=contract_address, abi=abi)

block_data = chain.eth.get_block(1)
transaction = chain.eth.get_transaction(block_data['transactions'][0])
print(dict(transaction))
print(1 , transaction['from'], transaction['input'])
```

HTML & CSS

explorer.html

```
<!DOCTYPE html>
<html>
<head>
<link href="/static/explorerDesign.css" rel="stylesheet" />
```

```
</head>
<body>
<nav class="nav-bar">
<form>
<button class="nav1">Explorer</button>
</form>
<form action="/wallet">
<button class="nav2">Wallet</button>
</form>
</nav>
<div class="blocks-container">
        {% block content %} {% endblock %}
</div>
</body>
</html>
```

blocks.html

```
{% extends 'explorer.html' %}
{% block content %}
<div class="block b0">
<div class="block-title">Block #0</div>
<div class="block-details">
<div>From : (None)</div>
<div>To: (None)</div>
<div>Value :None</div>
</div>
</div>
{% endblock %}
```

explorerDesign.css

```
body{
   margin: 0px;
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
```

```
.nav-bar{
   height: 48px;
   width: 100%;
   position: fixed;
   top: 0;
   background: #FFFFFF;
   box-shadow: 0px 2px 2pxrgba(0, 0, 0, 0.25);
   display: flex;
   align-items: center;
}

.nav1,.nav2{
   margin: 0px 20px 0px 20px;
}
.nav1{
   text-decoration: underline;
   font-weight: bold;
}
.nav2{
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
button{
   border: none;
   background: none;
}
.blocks-container{
   display: grid;
   justify-items: center;
   margin-top: 50px;
   height: 100%;
}
.block{
   display: block;
   border-radius: 15px;
   width: 60%;
   height: fit-content;
   min-height: 100px;
```

```
    background-color: #B7FFF6;
    margin: 10px 0px 10px 0px;
    overflow: hidden;
}

.block-title{
    font-weight: bold;
    font-size: large;
    margin: 15px 0px 10px 20px;
}

.block-details{
    display: flex;
    font-size: 14px;
    margin: 20px 30px 10px 20px;
    justify-content: space-between;
}
```

walletLogin.html

```
<!DOCTYPE html>
<html>
<head>
<link href="/static/walletLoginDesign.css" rel="stylesheet">
</head>
<body>
<nav class="nav-bar">
<form action="/">
<button class="text-button nav1">Explorer</button>
</form>
<form>
<button class="text-button nav2">Wallet</button>
</form>
</nav>
<div class="login-scaffold">
<div class="login-box">
```

```
<div class="login-title">
                    Login
</div>
<form class="login-form" action="/login">
<input name="username" placeholder="Username"><br>
<input name="password" type="password" placeholder="Password"><br>
<input class="login-button" type="submit">
</form>
<div class="re-direct">
<div>Don't have a wallet?</div>
<a class="text-button create-wallet-button" href="/create-account">Create
Wallet</a>
</div>
</div>
</div>
</body>
</html>
```

walletLoginDesign.css

```
body{
    margin: 0px;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.nav-bar{
    height: 48px;
    width: 100%;
    position: fixed;
    top: 0;
    background: #FFFFFF;
    box-shadow: 0px 2px 2pxrgba(0, 0, 0, 0.25);
    display: flex;
    align-items: center;
}

.nav1,.nav2{
    margin: 0px 20px 0px 20px;
```

```
}
.nav1{
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.nav2{
   text-decoration: underline;
   font-weight: bold;
}
.text-button{
   border: none;
   background: none;
}

.login-scaffold{
   display: flex;
   justify-content: center;
   margin-top: 60px;
}

.login-box{
   height: 340px;
   width: 310px;
   padding: 20px;
   border-radius: 20px;
   background-color: #B7FFF6;
   margin-top: 80px;
}

.login-title{
   font-size: 32px;
   font-weight: bold;
   margin: 30px 0px 40px 20px;
}

.login-form{
   display: grid;
   justify-content: center;
```

```
}

input{
   border: none;
   border-radius: 20px;
   height: 34px;
   width: 290px;
   padding: 10px;
   font-weight: bold;
   /*margin: 10px;*/
}

.login-button{
   border: none;
color: white;
   font-size: 20px;
   font-weight: bolder;
   background-color: black;
   border-radius: 20px;
   height: 54px;
   width: 310px;
}
.re-direct{
   display: flex;
   font-size: 12px;
   justify-content: space-between;
   margin: 10px 20px 5px 20px;
}

.create-wallet-button{
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
   font-weight: bold;
color: dodgerblue;
   text-decoration: underline;
}
```

createWallet.html

```
<!DOCTYPE html>
<html>
<head>
<link href="/static/createWalletDesign.css" rel="stylesheet">
</head>
<body>
<nav class="nav-bar">
<form action="/">
<button class="text-button nav1">Explorer</button>
</form>
<form>
<button class="text-button nav2">Wallet</button>
</form>
</nav>
<div class="signup-scaffold">
<div class="signup-box">
<div class="signup-title">
                    Create Account
</div>
<form class="signup-form" action="/signup">
<input name="username" placeholder="Username"><br>
<input name="password" type="password" placeholder="Password"><br>
<input name="retype-password" type="password" placeholder="Retype
password"><br>
<input class="signup-button" type="submit">
</form>
<div class="re-direct">
<div>Have a wallet?</div>
<a class="text-button login-button" href="/login-page">Login</a>
</div>
</div>
</div>
</body>
</html>
```

createWalletDesign.css

```
}

body{
   margin: 0px;
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.nav-bar{
   height: 48px;
   width: 100%;
   position: fixed;
   top: 0;
   background: #FFFFFF;
   box-shadow: 0px 2px 2pxrgba(0, 0, 0, 0.25);
   display: flex;
   align-items: center;
}

.nav1,.nav2{
   margin: 0px 20px 0px 20px;
}
.nav1{
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.nav2{
   text-decoration: underline;
   font-weight: bold;
}
.text-button{
   border: none;
   background: none;
}

.signup-scaffold{
   display: flex;
   justify-content: center;
   margin-top: 60px;
}
```

```
.signup-box{
   height: fit-content;
   width: 310px;
   padding: 20px;
   border-radius: 30px;
   background-color: #B7FFF6;
   margin-top: 80px;
}

.signup-title{
   font-size: 28px;
   font-weight: bold;
   margin: 30px 0px 40px 20px;
}

.signup-form{
   display: grid;
   justify-content: center;
}

input{
   border: none;
   border-radius: 20px;
   height: 34px;
   width: 290px;
   padding: 10px;
   font-weight: bold;
   /*margin: 10px;*/
}

.signup-button{
   border: none;
color: white;
   font-size: 20px;
   font-weight: bolder;
   background-color: black;
   border-radius: 20px;
   height: 54px;
   width: 310px;
}
.re-direct{
   display: flex;
   font-size: 12px;
   justify-content: space-between;
   margin: 10px 20px 5px 20px;
}

.login-button{
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
   font-weight: bold;
color: dodgerblue;
   text-decoration: underline;
}
```

wallet.html

```
<!DOCTYPE html>
<html>
<head>
<link href="/static/walletDesign.css" rel="stylesheet">
</head>
<body>
<nav class="nav-bar">
<form action="/">
<button class="text-button nav1">Explorer</button>
</form>
<form>
<button class="text-button nav2">Wallet</button>
</form>
</nav>
<div class="wallet-scaffold">
<div class="wallet-box">
<div class="account-name">
```

```
{{ name }}
</div>
<div class="account-address">
{{ address }}
</div>
<div class="account-balance">
{{ balance }}
</div>
<div class="token-symbol">
{{ symbol }}
</div>
<div class="send-box">
<div class="send-title">
                        Send
</div>
<form class="send-form" action="/send">
<input name="to" placeholder="To (address)"><br>
<input name="amount" placeholder="Value"><br>
<button class="send-button" type="submit">Send</button>
</form>
</div>
<div class="log-out">
<a class="text-button log-out-button" href="/login-page">Log Out</a>
</div>
</div>
</div>
</body>
</html>
```

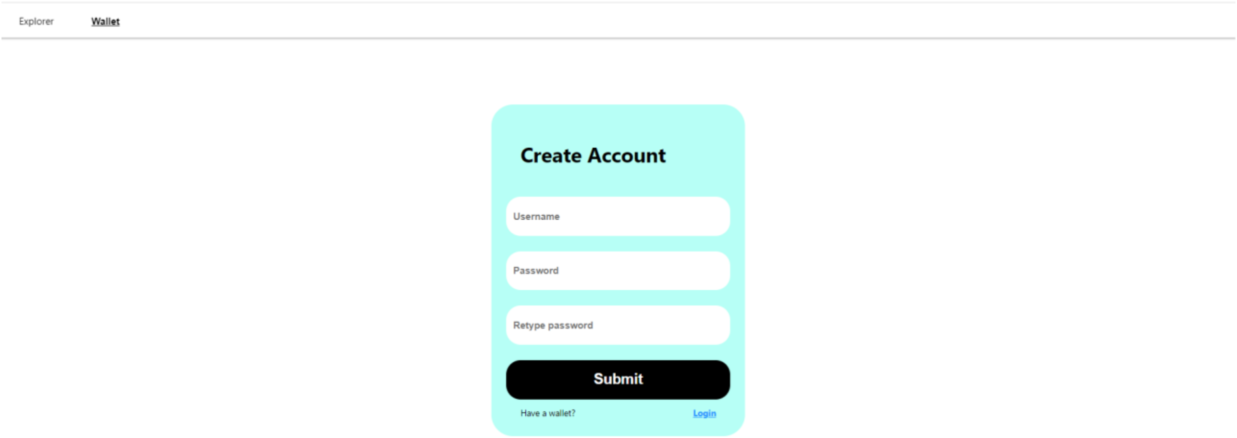
walletDesign.css

```
body{
    margin: 0px;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.nav-bar{
    height: 48px;
    width: 100%;
    position: fixed;
    top: 0;
    background: #FFFFFF;
    box-shadow: 0px 2px 2pxrgba(0, 0, 0, 0.25);
    display: flex;
    align-items: center;
}

.nav1,.nav2{
    margin: 0px 20px 0px 20px;
}
.nav1{
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.nav2{
    text-decoration: underline;
    font-weight: bold;
}
.text-button{
    border: none;
    background: none;
}

.wallet-scaffold{
    display: flex;
    justify-content: center;
    margin-top: 60px;
}

.wallet-box{
    display: grid;
    height: fit-content;
    width: fit-content;
    min-width : 310px;
    padding: 20px;
```

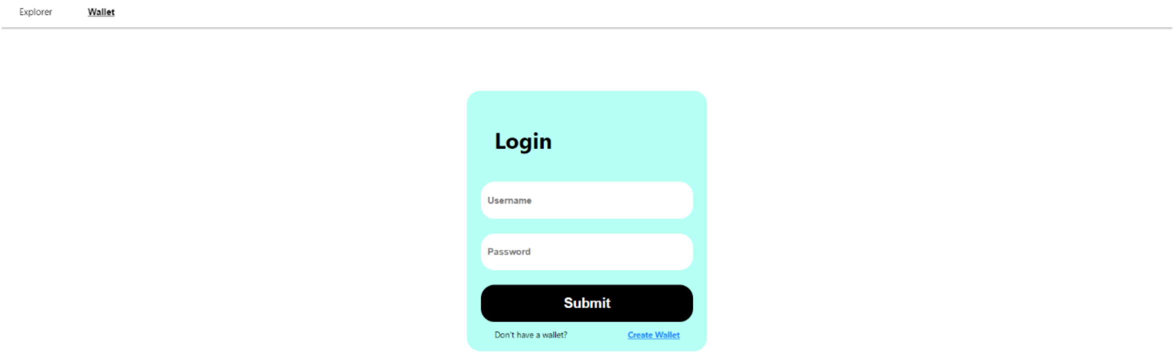
```
    border-radius: 30px;
    background-color: #B7FFF6;
    margin-top: 80px;
    justify-items: center;
}

.account-name{
    font-size: 20px;
    font-weight: 600;
    /*margin: 10px 0px 10px 0px*/;
}

.account-address{
    font-size: 12px;
    margin-bottom: 10px;
}
.account-balance{
    font-weight: 700;
    font-size: 24px;
}
.token-symbol{
    font-weight: 500;
    font-size: 20px;
}

.send-box{
    display: grid;
    border-radius: 25px;
    background-color: #51FFEA;
    margin: 20px 0px 10px 0px;
    padding: 20px 20px 10px 20px;
}

.send-title{
    font-weight: bold;
    font-size: 16px;
    margin: 0px 0px 10px 10px;
```

```
}

input{
   border: none;
   border-radius: 15px;
   height: 24px;
   width: 210px;
   padding: 10px;
   font-weight: bold;
   margin: 5px;
}

.send-button{
   border: none;
color: white;
   font-size: 16px;
   font-weight: bolder;
   background-color: black;
   border-radius: 18px;
   height: 44px;
   width: 220px;
   margin: 10px;
}

.log-out-button{
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
   font-weight: bold;
color: dodgerblue;
   text-decoration: underline;
}
```

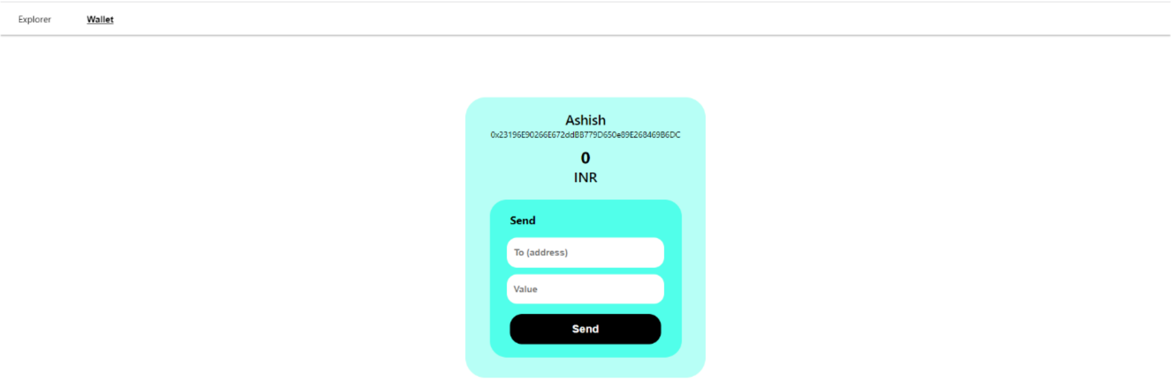
B. SYSTEM SNAPSHOTS



Creation of Wallet



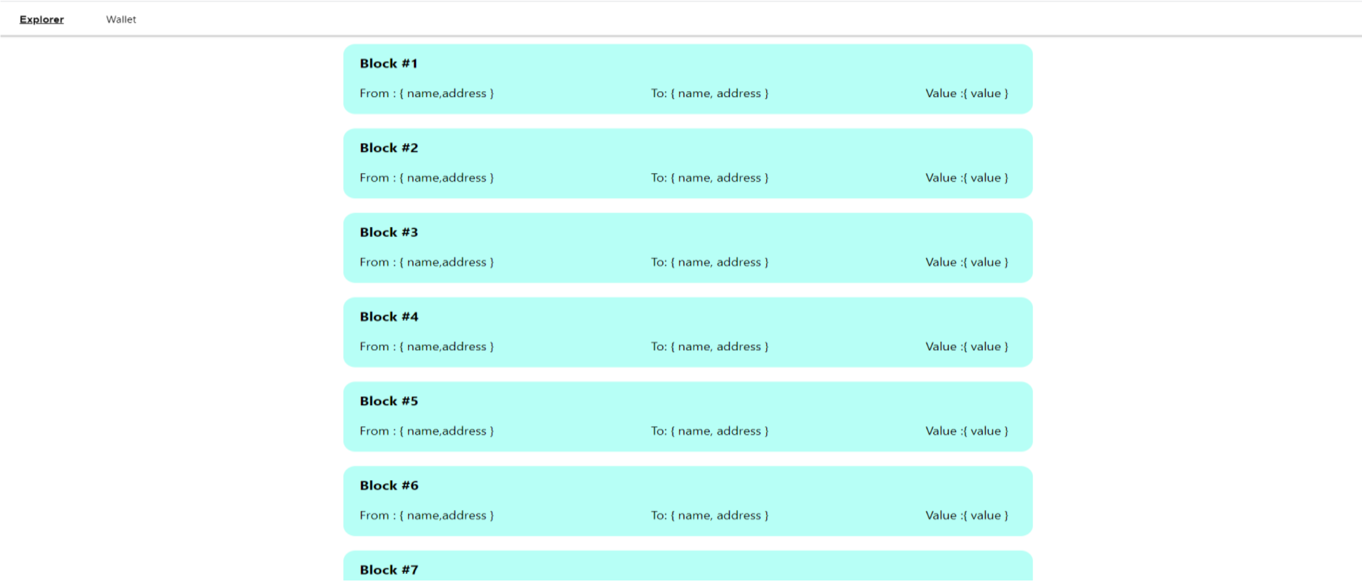
Login Page



Ashish
0x23196E90266E672ddBB779D650e89E268469B6DC

0
INR

Exchange Tokens Via Wallet



Block #1

From : { name,address }   To: { name, address }   Value :{ value }

Block #2

From : { name,address }   To: { name, address }   Value :{ value }

Block #3

From : { name,address }   To: { name, address }   Value :{ value }

Block #4

From : { name,address }   To: { name, address }   Value :{ value }

Block #5

From : { name,address }   To: { name, address }   Value :{ value }

Block #6

From : { name,address }   To: { name, address }   Value :{ value }

Block #7

Block Explorer that Can be visualized to the user

C. RESEARCH PAPER

TRANSPARENT BLOCKS

MUDUNURI SAI SIDDHARTHA VARMA[1]
STUDENT
DEPARTMENT OF CSE
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
CHENNAI
mudunurisai4656@gmail.com

TAMALALA VENKATA SIVA BHANU TEJA[2]
STUDENT
DEPARTMENT OF CSE
SATHYABAMA INSTITUTE OF SCIENCE AND TECNOLOGY
CHENNAI
bhanuteja981@gmail.com

MS. NANCY KIRUPANITHI.D [3]
ASSISTANT PROFESSOR
DEPARTMENT OF CSE
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
CHENNAI
nancykirupanithi.cse@sathyabama.ac.in

Abstr **eceiv**
money in the form of income tax, GST.etc. and as a common man, there is no way for us to know the amount of money they receive through taxes and also where the money is being spent. Also,The economics of national, state, and local government entities is impacted by corruption in India in a variety of ways. The causes of corruption are mainly due to a lack of transparency and traceability. The proposed blockchain project can bring transparency and traceability and helps people to see the amount of money being received and transferred by the government, hence reducing corruption.

I. INTRODUCTION

It is simpler to record transactions and track assets inside a company network thanks to blockchain, a distributed, immutable ledger. An asset might be material (such a home, automobile, sum of money, or plot of land) or intangible (intellectual property, patents, copyrights, branding). With the ability to monitor and trace almost anything of value, a blockchain network lowers risk and boosts efficiency for all stakeholders.

Commerce is driven by information. The better it is, the quicker it arrives, and the more accurate it is. Blockchain is the greatest technology for delivering such information since it provides instant, shareable, and completely transparent data that is held on an immutable ledger and accessible only by users of a permissioned network. Using a blockchain network, orders, payments, accounts, production, and much more can all be recorded. Additionally, because members share a single agreed knowledge of the truth, you can see every step of a transaction from beginning to end.

State governments are responsible for handling a vast array of duties. Numerous transactions are necessary for the state governments to carry out the numerous tasks that must be done at once. This comprises new projects, maintenance

ork
ker

Low-level corruption, which can be difficult to trace and hinders state growth, is a significant challenge for the top administration. The existing system makes it exceedingly difficult to track it.

Blockchain technology has the potential to significantly alter our corporate environment and will be extremely influential over the coming few decades. It has the potential to alter our economic structure and the way we see corporate operations. Blockchain is a distributed ledger that is decentralised and aims to provide transparency, data security, and integrity since it cannot be changed or manipulated. Only a small portion of current research on blockchain technology is geared toward examining its use in contexts or industries other than cryptocurrencies like Bitcoin. The majority of current research on blockchain technology is concentrated on its use in cryptocurrencies like Bitcoin.

Blockchain technology is more than simply bitcoin; it has a number of uses in business process management, government, banking, and finance. As a result, this study makes an effort to look into and examine the opportunities and difficulties associated with present and potential Blockchain Technology implementations. As a result, a sizable number of published papers were thoroughly examined and studied in light of their contributions to the field of Blockchain research.

The Indian government receives a significant amount of cash in the form of tax, GST, and other taxes, but as ordinary citizens, we have no means of knowing how much money they receive from taxes or how it is being spent. The economy of the municipal, state, and federal governments is also impacted in many ways by the corruption issue in India. The absence of transparency and traceability is a major contributor to corruption. The proposed blockchain initiative can increase transparency and traceability by allowing individuals to understand how much money the government receives and transfers, eliminating corruption. Transparent Blocks is a project where a token is created using smart contracts deployed on the Ethereum network and a block explorer which allows users or peers to see all the transactions that are made on the network to increase transparency.

II. Literature Survey

To combat certificate counterfeiting, a blockchain-enabled digital certificate system will be created. Blockchain immutability makes it possible to create digital certificates that are verifiable and anti-forgery. In this method, an online certificate is created in the manner described below. First, an electronic file containing the associated alternative connection data from the paper certificate is created, and its hash value is determined. In the chain system block within the block, a hash value is lastly saved. When a certificate is linked to a paper certificate, the system will generate a QR code and a string query key. The data required to confirm a physical certificate's authenticity can be found through an internet search or a phone scan.

Blockchain, sometimes referred to as the chain of blocks or a distributed ledger technology, may hold different transactions and operations without the aid of a trustworthy third-party program. Blockchain encourages integrity, accountability, and some degree of confidentiality since it is irreversible due to a variety of public and private keys. The popularity of blockchain has increased since the Bitcoin explosion. Key features of blockchain technology have been sought to be utilized in a number of applications and use cases. This document describes how blockchain technology is being used and some of the ways it is being used to safeguard and demonstrate the reliability of intelligent systems. Everyone who reads this white paper, in particular, will have a solid understanding of the use and applications of blockchain technology [2].

III. Existing System

At the moment, corruption in India is a problem that has a wide range of effects on the finances of the federal, state, and municipal governments. The economy of India has been stifled, according to the blame game. Excessive regulations, complex tax and licensing laws, a large number of government agencies, the monopoly of institutions under the control of the government over the delivery of specific goods and services, and opaque laws and procedures are some of the factors that contribute to corruption in India. Corruption activities can only be detected through Self-reporting, Citizen reporting, or Journalism and media reporting. Also, there is no way for a common man to see the amount of money the government of India receives through taxes and also where the money is being used except through the physical records which are maintained and are not accessible to everyone. The proposed project using blockchain can bring transparency and traceability and helps people to see the amount of money that is being received and transferred by the government, hence reducing corruption.

IV. Proposed System

n this system we are going to create and deploy a smart contract over the
Ethereum chain such that the smart contract is able to process, validate and

complete the transactions efficiently using Wallet to store and send
our tokens based on certain conditions, the system allows only verified users to
send and receive tokens, along with a user interface where the users can verify
their wallet address, and view all the transactions happening over the blockchain.
Thus increasing transparency and traceability.

Advantages:

[1] Less Block time

[2] Decentralization and Security

[3] Less gas fee

[4] Proof of Stake (PoS) consensus mechanism is better than Pow

[5] Transparency and traceability

This chapter discusses the goals and objectives of the project that will be accomplished after it is finished. These are the aims and objectives.:

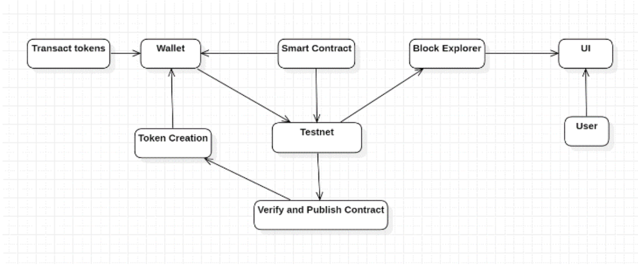
[1] Create a smart contract using solidity that can be deployed over the Ethereum network.

[2] Create a token that can be transacted using wallets such as Wallet

[3] Create a block explorer that reads the transactions and displays them.

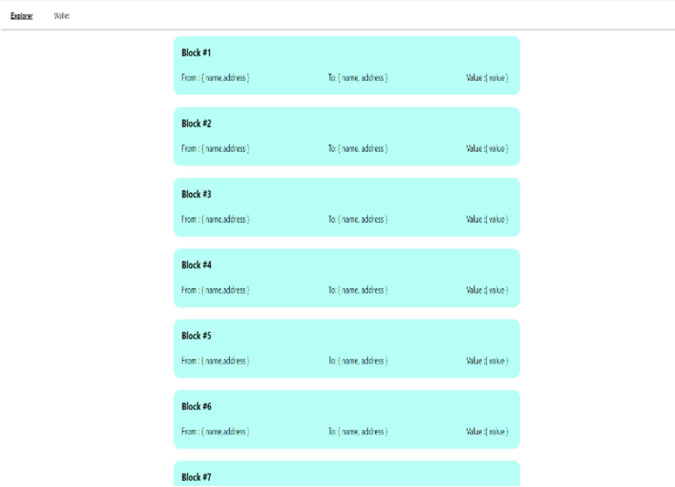
[4] Create a user interface that allows users to verify their wallet address and also to view all the transactions on the blockchain.

Fig.1.System Architecture for Transparent Blocks



V. RESULT

•



Block #1
From : { name,address } To: { name, address } Value :{ value }

Block #2
From : { name,address } To: { name, address } Value :{ value }

Block #3
From : { name,address } To: { name, address } Value :{ value }

Block #4
From : { name,address } To: { name, address } Value :{ value }

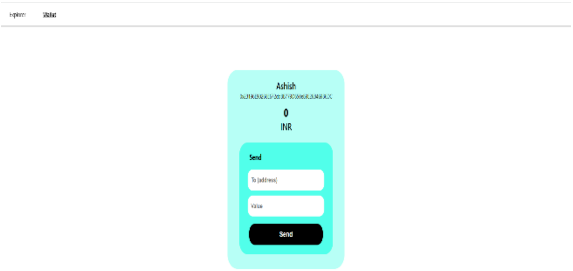
Block #5
From : { name,address } To: { name, address } Value :{ value }

Block #6
From : { name,address } To: { name, address } Value :{ value }

Block #7

Fig.2.Smart contract creation and deployment using Wallet on Ganache

The goal of the system testing method was to identify any flaw in our product. Several observations were observed when the software was subjected to a series of test inputs. Whether or not the software acts as predicted will be determined based on these observations. Two tiers of testing were conducted on our project.



Ashish
[illegible]
0
INR
Send
To (address)
Value
Send

When a module has been developed and successfully tested, unit testing is conducted. In order to test a single module, we must supply a complete environment, or everything else in addition to the module we would need. the processes from other modules that are called by the test module. a method for calling the test module's functions with the proper parameters. Accessible non-local data structures via the module. On each module that is detailed in the module description, unit testing was performed. In this kind of testing, we put the project module's different integrations to the test by giving it input. Testing the module interfaces is the main goal in order to make sure that no problems happen while invoking one module from another. The Smart contract is implemented, and the token is successfully produced.

.In this type of testing, we test various integration of the project module by providing input. Testing the module interfaces is the main goal in order to make sure that no problems happen while invoking one module from another.

VI. CONCLUSION AND FUTURE WORK

The Smart Contract is created using solidity to create a token such that the token can only be sentto a verified user iscompiled successfully and is deployed on the goerli Ethereum testnet and Wallet is used as a wallet to store and transfer the token. And a user interface is created on the web using python, flask, Html, Css which allows users to verify their account so that they can start receiving and transferring the token, search the wallet addresses of the users based on name or Id, and view all the transactions on the blockchain.

ACKNOWLEDGMENT

I wish to thank the professors at Sathyabama University for allowing me to perform this study. We would particularly like to thank them for her help. MS. NANCYKIRUPANITHI.D

REFERENCES

[1] Mohanta, B. K., Panda, S. S., & Jena, D. (2018, July). An overview of smart contracts and use cases in blockchain technology. In 2018 9th international conference on computing, communication and networking technologies (ICCCNT). IEEE.

[2] Pandey, S., Ojha, G., Shrestha, B., & Kumar, R. (2019, May). Block SIM: A practical simulation tool for optimal network design, stability, and planning. In 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE.

[3] Ehmke, C., Wessling, F., & Friedrich, C. M. (2018, May). Proof-of-property: a lightweight and scalable blockchain protocol. In Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain.

[4] Caldarelli, G., & Ellul, J. (2021). Trusted academic transcripts on the blockchain: A systematic literature review. Applied Sciences, 11(4), 1842.

[5] Alam, A. (2022). Platform Utilising Blockchain Technology for eLearning and Online Education for Open Sharing of Academic Proficiency and Progress Records. In Smart Data Intelligence (pp. 307-320). Springer, Singapore.

[6] Caldarelli, G., & Ellul, J. (2021). Trusted academic transcripts on the blockchain: A systematic literature review. Applied Sciences, 11(4), 1842.

[7] Grover, P., Kar, A. K., & Janssen, M. (2019). Diffusion of blockchain technology: Insights from academic literature and social media analytics. Journal of Enterprise Information Management.

[8] Massaro, Maurizio. "Digital transformation in the healthcare sector through blockchain technology. Insights from academic research and business developments." Technovation (2021): 102386.

[9] Zou, Y., Meng, T., Zhang, P., Zhang, W., & Li, H. (2020). Focus on blockchain: A comprehensive survey on academic and application. IEEE Access, 8, 187182-187201.

[10] Janowicz, K., Regalia, B., Hitzler, P., Mai, G., Delbecque, S., Fröhlich, M., ... & Lazarus, T. (2018). On the prospects of blockchain and distributed ledger technologies for open science and academic publishing. Semantic web, 9(5), 545-555.

[11] Halpin, H., &Piekarska, M. (2017, April). Introduction to Security and Privacy on the Blockchain. In 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 1-3). IEEE.

[12] Alharby, M., Aldweesh, A., & Van Moorsel, A. (2018, November). Blockchain-based smart contracts: A systematic mapping study of academic research (2018). In 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB) (pp. 1-6). IEEE.

[13] Mohan, V. (2019). On the use of blockchain-based mechanisms to tackle academic misconduct. Research Policy, 48(9), 103805.

[14] Jiang, S.; Cao, J.; Wu, H.; Yang, Y. Fairness-based Packing of Industrial IoT Data in Permissioned Blockchains. IEEE Trans. Ind. Inform. 2020, 1.

[15] Butijn, B.-J.; Tamburri, D.A.; Heuvel, W.-J.V.D. Blockchains. ACM Comput. Surv. 2020, 53, 1–37.