CREATING A TOKEN USING BLOCKCHAIN TECHNOLOGY

Submitted in partial fulfillment of the requirements for the award

of

Bachelor of Engineering degree in Computer Science and Engineering

By

V.N.V.V.S.PRAMODH (Reg.No – 39111081)

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 | 12B Status by UGC | Approved by AICTE

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 **V.N.V.V.S PRAMODH (39111081)** who carried out the Project Phase-2 entitled **"CREATING A TOKEN USING BLOCKCHAIN TECHNOLOGY"** under my supervision from Jan 2023 to April 2023.

Internal Guide
Ms.D.DEEPA.,M.E.,(Ph.D)

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, **V.N.V.V.S.PRAMODH (Reg.No – 39111081),** hereby declare that the Project Phase-2 Report entitled **CREATING A TOKEN USING BLOCKCHAIN TECHNOLOGY"** done by me under the guidance of **Ms.D.DEEPA.,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: 20-04-2023

PLACE: CHENNAI **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, and **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.D.Deepa.,M.E.,(Ph.D).,** for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my phase-2 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

Blockchain technology has enabled a new software paradigm for managing digital ownership in partial- or zero-trust environments. It uses tokens to conduct transactions, exchange verifiable data, and achieve coordination across organizations and on the web. Fundamental to this representation is that users can independently control token custody in digital wallets through public-key cryptography and interact with one another in a peer-to-peer manner. Blockchain networks provide secure transaction reconciliation, linkage, and storage in consolidated, integrity-protected distributed ledgers forming mutually operated record-keeping execution environments. Data models with varied capabilities and scopes have been defined to issue tokens, which additional protocols can help manage while enabling separation of concerns. Security and recovery mechanisms allow users to set up self-hosted, externally hosted, and hybrid account custody models. Scaling schemes have been developed to accommodate transactions off-chain with deferred on-chain settlement, as well as deposit contracts with built-in, self-enforceable conditions to exchange tokens without intermediaries, transaction submission rules to fit in with different deployment scenarios, and privacy-enhancing techniques to protect user confidentiality. Software design patterns and infrastructure tools can also make it easier to integrate blockchain networks, wallets, and external resources in user interfaces. This document provides a high-level technical overview and conceptual framework of token designs and management methods. It is built around five views: the token view, wallet view,transaction view, user interface view, and protocol view. The purpose is to lower the barriers to study, prototype, and integrate token-related standards and protocols by helping readers understand the building blocks involved both on-chain and off-chain.With This work provides a systematic literature review of blockchain-based applications across multiple domains. The aim is to investigate the current state of blockchain technology and its applications and to highlight how specific characteristics of this disruptive technology can revolutionize “business-as- usual” practices.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Bitcoin and Ethereum introduced the technologies of blockchains and smart contracts as well as new types of global, web-native social constructs with decentralized governance. Anyone with an internet connection can view the blockchain, operate a publishing node, and submit transactions. Blockchain addresses are derived from public keys generated directly by the users who control the associated private keys' custody. Transactions are signed using these private keys before being validated and reconciled by the blockchain nodes, providing data integrity and public verifiability.

More generally, the domain of permissionless blockchains brought about protocol-native tokens, or cryptocurrencies. With the potential to provide alternatives to existing institutions and market discipline due to their decentralized and non-sovereign nature, cryptocurrencies could have long-term implications for financial inclusion and stability . Moreover, permissionless blockchains brought about tamper-evident and tamper-resistant computing platforms that maintain a common state for all participants. Those platforms allow smart contract execution with application-specific instruction sets, such as Bitcoin, or general-purpose execution environments, such as the Ethereum. Smart contracts make it possible to integrate off-chain schemes, deploytokens, and more generally build blockchain applications.Tokens represent digital assets and serve as instruments for exchanging verifiable data. Fungible tokens are meant to be completely interchangeable, acting as digitalcoins and enabling payment systems. When they are native to a protocol and used to decentralize its governance, they are also largely but inconsistently referred toas platform or utility tokens. Tokens associated with unique identifiers, nonfungible tokens and stateful tokens, are meant to identify things or data uniquely. They can be part of wider supply chain or traceability frameworks. Blockchain-based services have emerged to help manage account custody and individual token ownership as well as to increase transaction throughput, protect user privacy, and provide infrastructure tools. **Problem Definition**

Creating a crypto token using Blockchain Technology and adding a Smart contract.

1.1 Scope and Objectives

- Research and select the appropriate blockchain platform for token creation, such as Ethereum, Binance Smart Chain, or Solana.
- Design and develop the cryptocurrency token using the selected blockchain platform and token standards, such as ERC-20, BEP-20, or SPL.
- Implement smart contracts to manage the creation, transfer, and management of the token.
- Ensure proper security measures, such as multi-factor authentication, encryption, and regular audits, are in place to ensure the safety of the token and its users.
- Develop marketing strategies to attract investors and increase adoption of the token.

Objectives:

- To create a cryptocurrency token that can be used as a means of exchange or investment.
- To establish a community around the token, increase its adoption, and drive its growth.
- To provide investors with a secure and transparent investment opportunity with clear tokenomics and a well-defined roadmap.
- To enable the token to be easily traded on decentralized exchanges, allowing for liquidity and price discovery.
- To explore potential use cases for the token beyond investment, such as in decentralized applications or as a means of payment.

CHAPTER 2

LITERATURE SURVEY

"Blockchain-Based Crypto Tokens: A User's Guide" by William Mougayar (2016)

This book provides a comprehensive guide to blockchain-based crypto tokens, including how they work, how to create them, and their use cases. It also covers the basics of blockchain technology and the benefits and drawbacks of using crypto tokens.

"Token Economics: The Value of Crypto Tokens" by Vladislav Martynov and Dominik Zynis (2018)

This paper discusses the economics of crypto tokens and how they can be used to incentivize specific behaviors. It covers various token models, including utilitytokens, security tokens, and payment tokens, and how they differ from traditional forms of currency.

"The Rise of Crypto Securities: What They Are, How They Work, and What They Mean for the Future" by Ryan Selkis (2018)

This article discusses the emergence of crypto securities and how they are different from traditional securities. It also covers the regulatory issues surrounding crypto securities and their potential impact on the financial industry.

"Smart Contract-Based Token Creation and Deployment on the Ethereum Blockchain" by Anthony Di Iorio and Vitalik Buterin (2018)

This paper provides a step-by-step guide to creating and deploying a smart contract-based token on the Ethereum blockchain. It covers the basics of Ethereum, including the Solidity programming language and the Ethereum Virtual Machine (EVM).

"Crypto Tokens and the Coming Age of Protocol Innovation" by Chris Dixon (2017)
This article discusses the potential for innovation in blockchain protocols using

crypto tokens. It covers the concept of "fat protocols" and how they differ from traditional internet protocols, as well as the potential for new types of applications to emerge using crypto tokens.

2.1 INFERENCES FROM LITREATURE SURVEY

Crypto tokens are a relatively new concept that has emerged as a result of the blockchain revolution. They are digital assets that can be used for various purposes, including incentivizing specific behaviors, representing ownership in a particular asset, or serving as a form of currency.

The creation of a crypto token requires a deep understanding of both blockchain technology and economics. Different token models, such as utility tokens, security tokens, and payment tokens, have different use cases, benefits, and drawbacks.

Smart contracts, which are self-executing agreements with the terms of the agreement between buyer and seller being directly written into lines of code, are a crucial component of creating a crypto token. Smart contracts enable the creation of rules and incentives that are transparent, secure, and automated.

Ethereum is currently the most widely used blockchain platform for creating crypto tokens. Solidity is the most common programming language used to create smart contracts on the Ethereum platform.

The emergence of crypto securities has raised regulatory concerns, as they are different from traditional securities and require a new regulatory framework.

2.2 Motivation

Customized tokens can be used to incentivize specific behaviors, such as completing tasks or contributing to a community, in a transparent and secure way. By creating a crypto token, a company or community can incentivize its members to act in a particular way, leading to increased engagement and participation.

Crypto tokens can represent ownership in a particular asset, such as real estate or artwork. By creating a token that represents ownership in a particular asset, the asset can be divided into smaller pieces, enabling more people to invest in it. This can democratize access to valuable assets and increase liquidity.

Crypto tokens can be used as a form of currency, either as a standalone currency or as a means of exchange within a particular community or platform. By creating a crypto token, a company or community can create its own currency, enabling more control over the monetary system and potentially reducing transaction fees.

Creating a crypto token using blockchain technology can provide a deeper understanding of both blockchain technology and economics, which are becoming increasingly important in today's digital economy. This knowledge can be valuable for individuals and businesses looking to participate in the growing blockchain ecosystem.

With the rise of blockchain technology and the increasing adoption of cryptocurrencies, there is a growing demand for customized tokens. By creating a crypto token, an individual or company can tap into this growing market and potentially generate revenue through the sale or use of the token.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

Lack of standardization: There are currently no industry-wide standards for creating crypto tokens, which can lead to inconsistencies in token models, smart contract code, and other aspects. This lack of standardization can lead to confusion, inefficiencies, and potential security risks.

Scalability issues: Blockchain technology is still in its early stages and faces significant scalability challenges, particularly in terms of transaction throughput and speed. These challenges can limit the number of transactions that can be processed on the blockchain, which can impact the usability of crypto tokens.

Security risks: Although blockchain technology is considered to be secure, there are still security risks associated with creating and using crypto tokens. For example, smart contract vulnerabilities, exchange hacks, and phishing attacks can all result in the loss of tokens or other assets.

Regulatory challenges: The regulatory landscape for crypto tokens is still evolving and can vary significantly from country to country. This can lead to uncertainty for businesses and individuals looking to create or use crypto tokens and can impact the growth and adoption of the technology.

Lack of user education: Despite the growing popularity of cryptocurrencies and blockchain technology, many users still lack a deep understanding of how they work and the potential risks and benefits. This can lead to misinformed decisions, such as investing in risky tokens or falling victim to scams.

Overall, these open problems highlight the need for continued research and development in the area of blockchain technology and crypto tokens. Addressing these challenges can help to improve the usability, security, and scalability of the technology, and ultimately drive its adoption and growth.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES / RISK ANALYSIS OF THE PROJECT

A risk analysis of the project title "Creating a Crypto Token using Blockchain Technology" involves identifying and assessing potential risks and developing strategies to mitigate them. Some potential risks are:Security risks, such as smart contract vulnerabilities, exchange hacks, and phishing attacks.Legal and regulatory risks, such as non-compliance with anti-money laundering and securities laws.Market risks, such as low demand for the token or competition from other tokens.Technology risks, such as blockchain platform instability, scalability limitations, and interoperability issues.Financial risks, such as market volatility and capital loss.To mitigate these risks, the project team should implement security measures such as code reviews and smart contract audits, ensure compliance with relevant regulations, conduct market research and analysis, and monitor the blockchain platform for potential issues. Additionally, the team should establish clear contingency plans and risk management strategies to address any unforeseen challenges that may arise during the development and deployment of the token.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware Requirements

- Computer: A computer with a minimum of 8 GB of RAM and a modern processor (such as an Intel Core i5 or equivalent) is recommended for smart contract development and blockchain node operation.
- Storage: The blockchain platform and development tools can consume significant amounts of storage space, so a computer with a minimum of 256 GB of storage (preferably SSD) is recommended.
- Graphics Processing Unit (GPU): A dedicated GPU may be required for certain blockchain platforms, such as Ethereum, to support mining or other computational tasks.
- Mobile Devices: If the project requires mobile wallet support, it may be

necessary to test the application on a range of mobile devices to ensure compatibility.

Software Requirements:

- Operating System **:** Windows 10, Kali Linux
- Language **:** Python 3
- Framework **:** Node Js

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small-Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although

Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

FirstPythonProgram

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

```
$ python

Python 2.4.3(#1, Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>
```

Type the following text at the Python prompt and press the Enter −

```
>>>print "Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!").** However, in Python version 2.4.3, this produces the following result −

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file −

```
Print "Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

```
$ python test.py
```

This produces the following result −

```
Hello, Python!
```

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The proposed system for creating a crypto token using blockchain technology is a decentralized application (DApp) that enables the creation and management of a custom cryptocurrency token based on blockchain technology. The system is designed to provide a secure, transparent, and immutable ledger for recording all transactions related to the token.

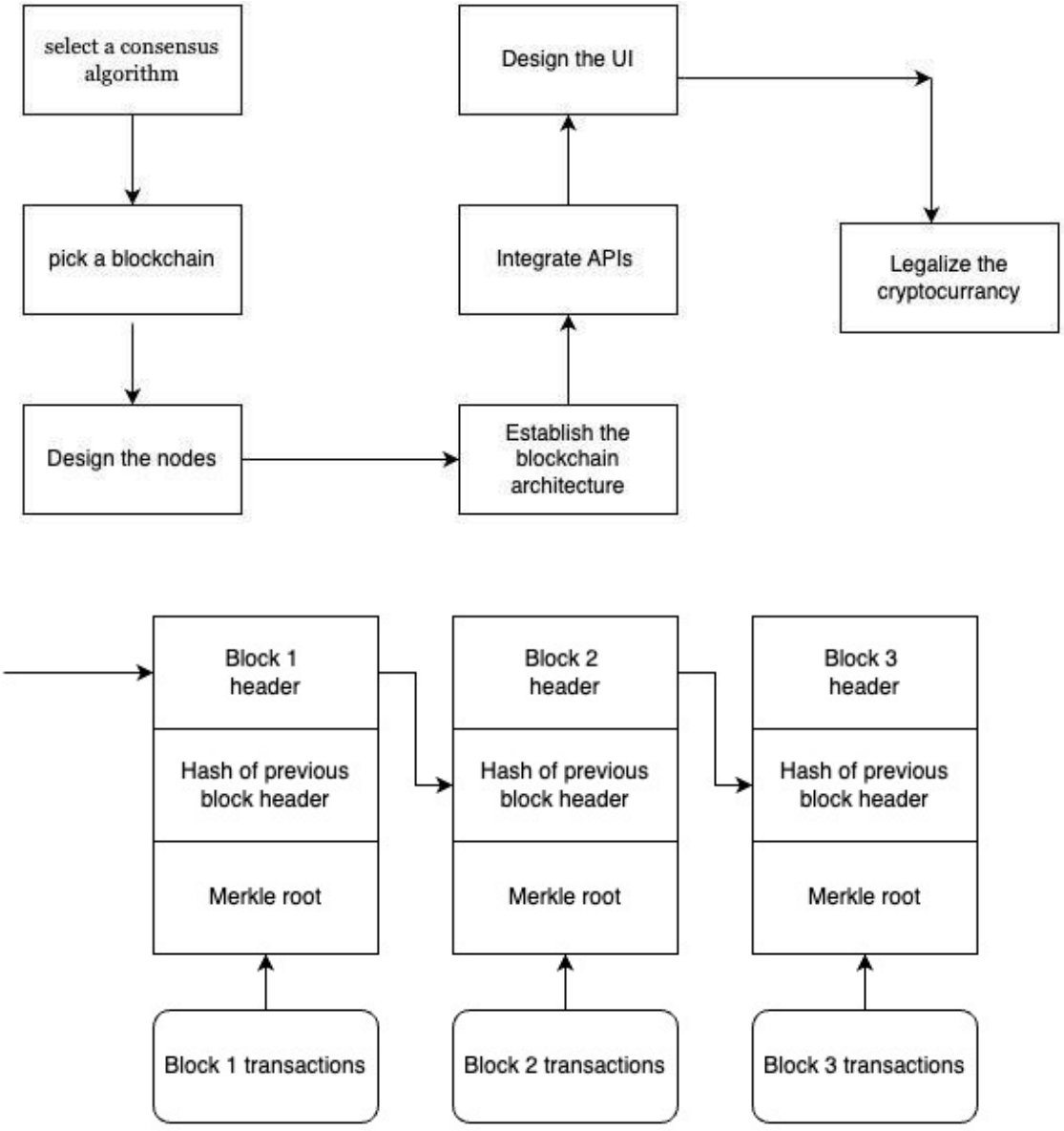
The system will consist of various components, including a smart contract that will govern the issuance and distribution of the token, a user interface for managing the token, and a blockchain network that will serve as the underlying infrastructure for the system. The smart contract will define the rules and conditions for the creation, distribution, and transfer of the token, while the user interface will provide users with a way to interact with the token.

The blockchain network will be responsible for validating and recording all transactions related to the token, providing a secure and tamper-proof record of all transactions. The system will leverage existing blockchain technologies, such as Ethereum or Binance Smart Chain, to create the token and manage its distribution.

Once the token is created, it can be used for various purposes, such as a medium of exchange, a store of value, or a tool for incentivizing specific behaviors. The system will enable the token issuer to define the specific use cases and applications of the token, providing a flexible and customizable tool for achieving their goals.

Overall, the proposed system for creating a crypto token using blockchain technology represents an innovative and powerful tool for enabling new forms of value exchange and incentivization, while leveraging the security and transparency of blockchain technology to ensure the integrity of all transactions.

4.2 ARCHITECTURE :



4.3 SELECTED METHODOLOGY

Methodology

Creating a crypto token using blockchain technology is a project that involves the development of a digital currency that is decentralized, secure, and immutable.

The main objective of the project is to create a blockchain-based token that can be used for various purposes such as investment, transactions, and as a store of value. The methodology for this project can be divided into six main stages, which are as follows:

- Research and Planning: The first stage of the project involves conducting thorough research on blockchain technology and its use cases. The team should also research existing crypto tokens, their functionalities, and their adoption rate. Based on the research, the team should then create a detailed project plan that outlines the project scope, timelines, budget, and resources required.
- Token Design: The second stage involves designing the token. The team should decide on the purpose of the token, its name, symbol, and the number of tokens to be issued. The team should also consider the token's characteristics such as its supply limit, distribution model, and whether it is mineable or not.
- Blockchain Development: The third stage involves developing the blockchain network on which the token will be built. The team should choose the appropriate blockchain platform and tools that best suit the project requirements. Ethereum is a popular blockchain platform for developing crypto tokens due to its smart contract functionality. The team should also develop and test the smart contracts that will govern the token's functionalities such as issuance, transfer, and storage.
- Token Development: The fourth stage involves developing the token using the blockchain network. The team should create and deploy the token contract on the blockchain platform, and conduct thorough testing to ensure that the token functions as intended. The team should also develop a user-friendly wallet application that enables users to store, transfer, and manage the token.
- Marketing and Adoption: The fifth stage involves marketing the token and creating awareness among potential investors and users. The team should create a white paper that outlines the token's purpose, technology, and potential benefits. The team should also establish partnerships with other blockchain projects and companies to increase the token's adoption rate.

- Maintenance and Upgrades: The final stage involves maintaining and upgrading the token and the blockchain network. The team should monitor the token's performance and address any issues that may arise. The team should also upgrade the token's functionalities as required, based on user feedback and market demand.

In conclusion, creating a crypto token using blockchain technology is a complex project that requires thorough research, planning, and execution. The project's success relies on the team's expertise in blockchain development, marketing, and user adoption. By following the above methodology, the team can increase the likelihood of creating a successful and sustainable blockchain-based token.

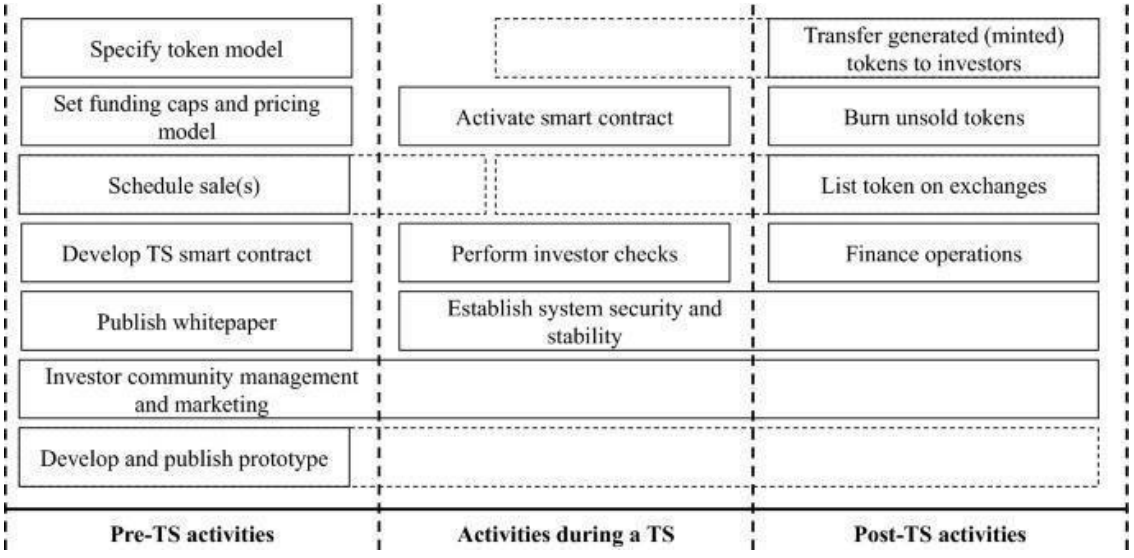


Table 2: Token Representation Types

	Blockchain-Native (Base Layer)	On Top of an Existing Blockchain (Smart Contract Layer)
UTXO-Based	System account balances are encoded as the sums of unspent transaction outputs of past transactions. Spending a token results in new, unspent transaction outputs. For example, bitcoin is Bitcoin protocol's native token.	A separate protocol, sometimes called *colored coin* method, encodes custom account balances or unique identifiers into extra metadata included in unspent transaction outputs of past transactions.
Account-Based	Variables in the blockchain's global state store system account balances assigned to blockchain addresses. For example, ether is Ethereum protocol's native token.	Variables in the blockchain's global state store custom account balances or unique identifiers assigned to blockchain addresses either centrally, within *token factory contracts*, or at the account level (i.e., data values and code are decoupled).

PROPOSED ALGORITHM :

Creating a crypto token using blockchain technology involves the use of various algorithms to ensure that the token is secure, decentralized, and immutable. The algorithms used in the project can be broadly classified into five categories, which are consensus algorithms, smart contract programming languages, hashing algorithms, cryptographic algorithms, and tokenomics algorithms.

4.6.1 Concensus Algorithms

Consensus algorithms are used to ensure that transactions are validated and added to the blockchain in a decentralized and secure manner. There are several consensus algorithms used in blockchain technology, including Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), and Proof of Authority (PoA).

Proof of Work (PoW):

The PoW algorithm is used in blockchain networks such as Bitcoin and Litecoin. It requires miners to perform complex mathematical calculations to validate transactions and add new blocks to the blockchain. The first miner to solve the mathematical problem is rewarded with new tokens. PoW is energy-intensive and requires significant computational power, making it less efficient than other consensus algorithms.

Proof of Stake (PoS):

The PoS algorithm is an alternative to the PoW algorithm and is used in blockchain networks such as Ethereum. It requires validators to hold a certain amount of tokens as a stake to validate transactions and add new blocks to the blockchain. Validators are selected based on their stake and are rewarded with new tokens. PoS is more energy-efficient than PoW, making it a popular consensus algorithm in the blockchain community.

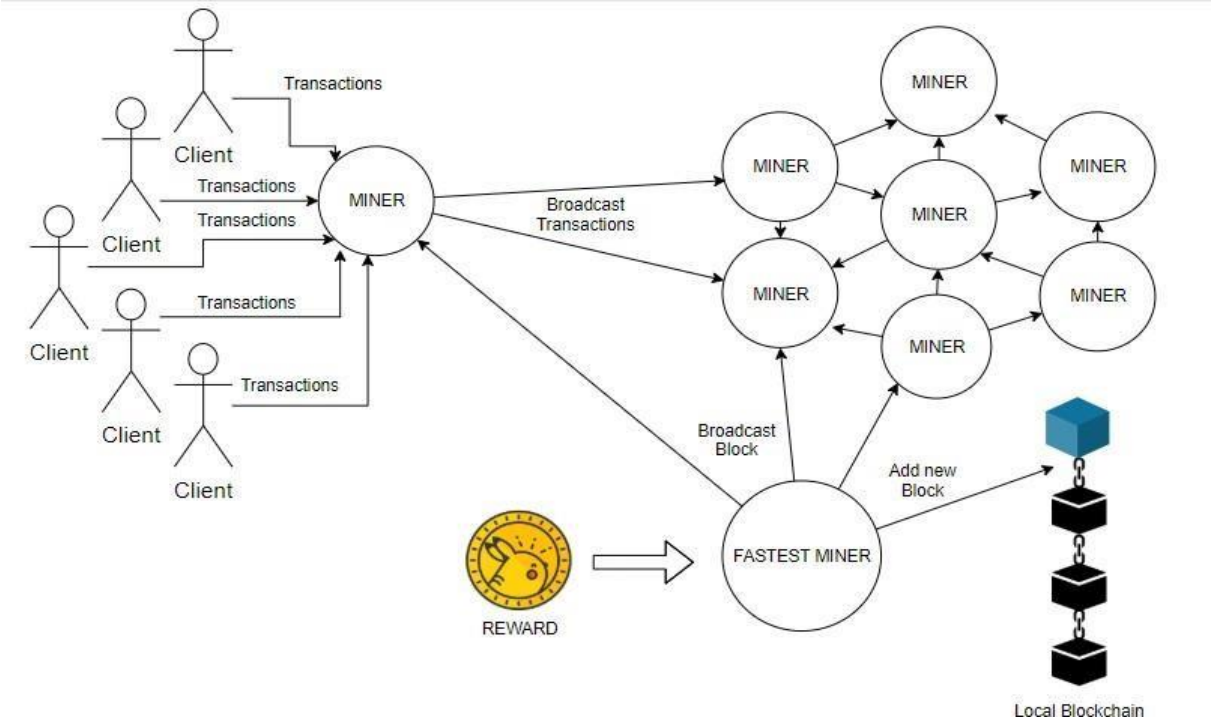
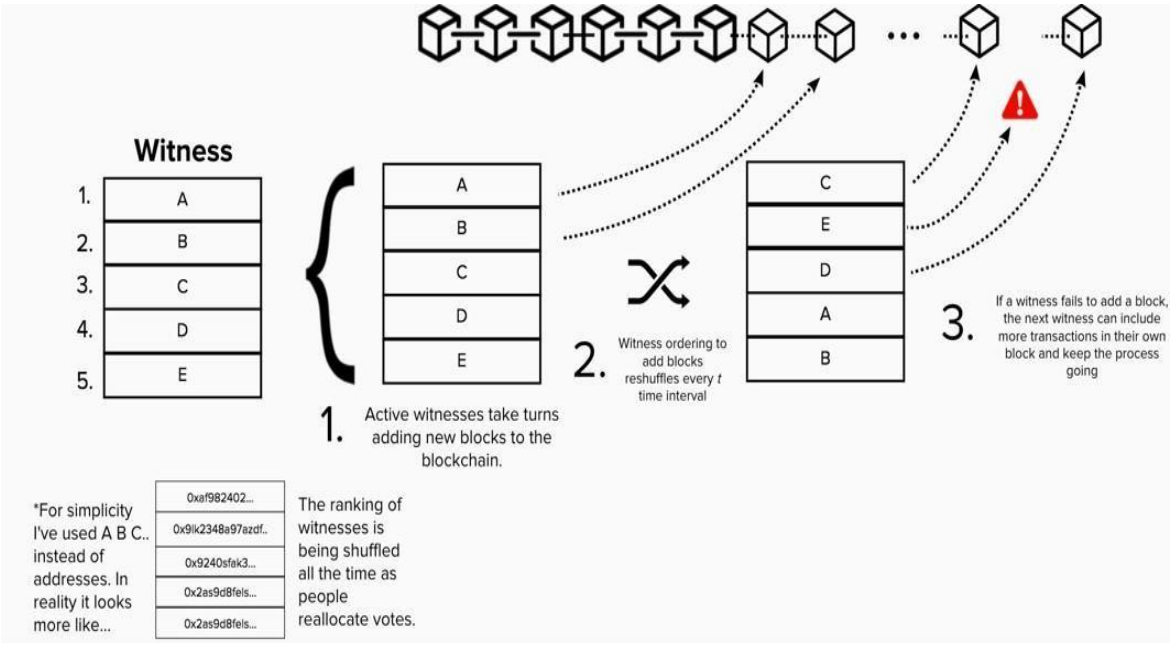


Fig . 4.2.1. Pow

Delegated Proof of Stake (DPoS):

The DPoS algorithm is used in blockchain networks such as EOS and Tron. It requires token holders to vote for delegates who will validate transactions and add new blocks to the blockchain. Delegates are selected based on the number of votes they receive from token holders. DPoS is more energy-efficient than PoW and PoS, making it a popular consensus algorithm in blockchain networks that require fast transaction processing.



Proof of Authority (PoA):

The PoA algorithm is used in private blockchain networks that require fast transaction processing and low energy consumption. It requires a group of pre-approved validators to validate transactions and add new blocks to the blockchain. Validators are selected based on their reputation and trustworthiness. PoA is a highly centralized consensus algorithm, making it less secure than other consensus algorithms in the blockchain community.

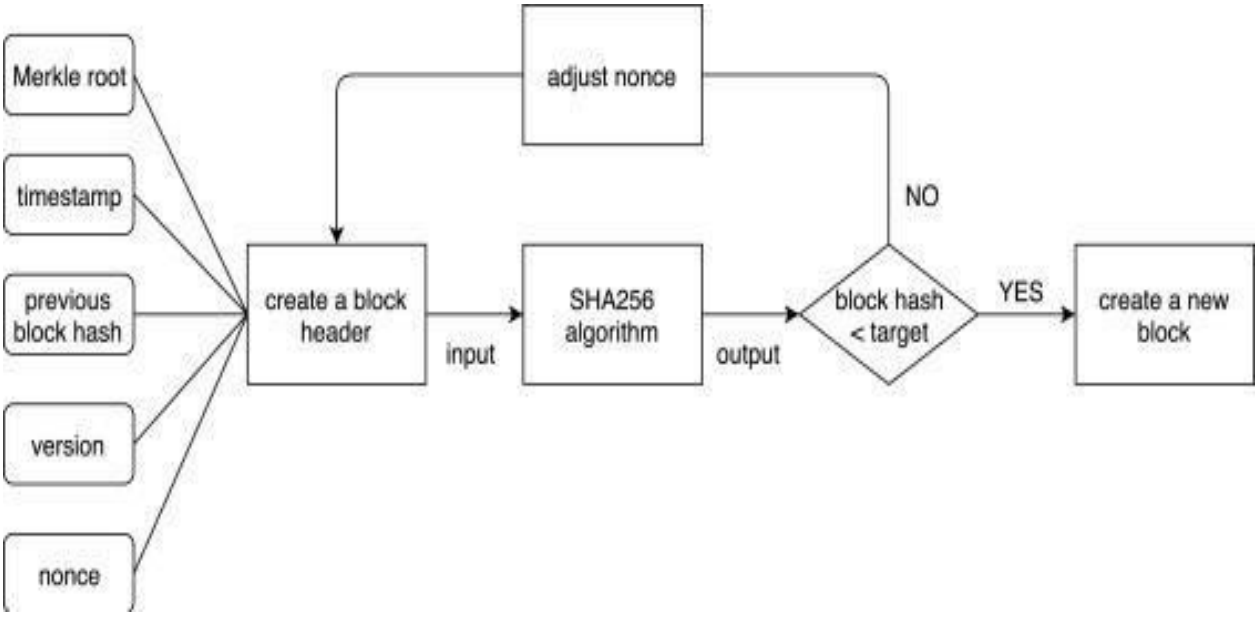


Fig . 4.2.1 Protocols of using consensus Algorithm

4.6.2 Smart contract Algorithms

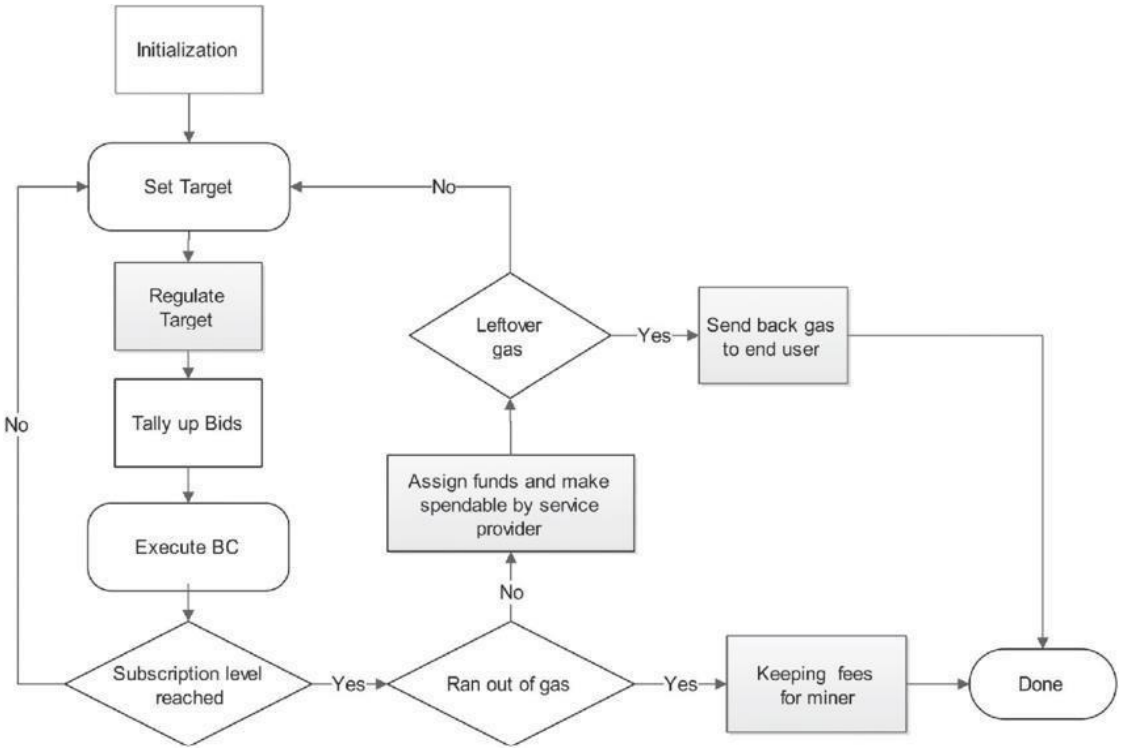
Smart contracts are self-executing contracts that are programmed on the blockchain. They are used to define the rules and conditions of the contract and execute them automatically when triggered. There are several programming languages used to develop smart contracts, including Solidity, Vyper, and Rust.

Solidity: Solidity is a popular programming language used to develop smart contracts on the Ethereum blockchain. It is a high-level language that is similar to JavaScript and is used to define the rules and conditions of the contract. Solidity allows developers to create custom tokens, set up token distribution, and implement token burning, among other functionalities.

Vyper: Vyper is a programming language used to develop smart contracts on the Ethereum blockchain. It is a high-level language that is designed to be more secure and easier to audit than Solidity. Vyper does not allow for some of the advanced features available in Solidity, making it less versatile but more secure.

Rust: Rust is a programming language used to develop smart contracts on the NEAR blockchain. It is a low-level language that is designed to be more secure and efficient than other programming languages. Rust allows developers to write complex smart contracts that can execute complex transactions and implement complex logic.

Hashing Algorithms: Hashing algorithms are used to encrypt data on the blockchain. They are used to generate a unique, fixed-length hash value that is used to verify the integrity of the data on the blockch


Initialization
Set Target
No
Regulate Target
Leftover gas
Yes
Send back gas to end user
No
Tally up Bids
Assign funds and make spendable by service provider
Execute BC
No
Subscription level reached
Yes
Ran out of gas
Yes
Keeping fees for miner
Done

CHAPTER 5
SYSTEM DESIGN

The system design for creating a crypto token using blockchain technology involves several components, including the blockchain network, the smart contract, the token contract, and the user interface. Each component plays a critical role in ensuring the security, decentralization, and functionality of the crypto token.

Blockchain Network:
The first component of the system design is the blockchain network. There are several blockchain networks available, including Ethereum, Binance Smart Chain, and Polygon. The choice of blockchain network will depend on factors such as transaction processing speed, security, and cost. Once the blockchain network is chosen, a node will be set up to interact with the network.

Smart Contract:
The second component of the system design is the smart contract. The smart contract is a self-executing contract that is programmed on the blockchain. It is used to define the rules and conditions of the contract and execute them automatically when triggered. The smart contract will be developed using a smart contract programming language such as Solidity, Vyper, or Rust. The smart contract will contain the logic for creating and managing the token.

Token Contract:
The third component of the system design is the token contract. The token contract is a smart contract that is used to create and manage the token. It will define the name, symbol, total supply, and decimal places of the token. It will also contain functions for minting new tokens, transferring tokens between addresses, and burning tokens. The token contract will interact with the smart contract to ensure that the token functions properly.

User Interface:
The fourth component of the system design is the user interface. The user interface is the front-end component of the application that allows users to interact with the token. It will be developed using a web development framework such as React or Vue.js. The user interface will allow users to view their token balance, transfer tokens, and interact with the smart contract.

Overall, the system design for creating a crypto token using blockchain technology involves setting up a blockchain network, developing a smart contract using a smart contract programming language, creating a token contract to manage the token, and developing a user interface to allow users to interact with the token. Each component is critical to ensuring the security, decentralization, and functionality of the crypto token.

5.1 RESEARCH OBJECTIVE :

The research objective for the project of creating a crypto token using blockchain technology is to explore the use of blockchain technology and smart contracts in creating a decentralized digital asset. The specific research objectives include:

Understanding the fundamentals of blockchain technology and smart contracts:
The project aims to gain a deep understanding of the underlying principles of blockchain technology and smart contracts. This includes understanding the benefits and limitations of using blockchain technology for creating decentralized digital assets.

Exploring different blockchain networks:
There are several blockchain networks available, each with their own unique features and capabilities. The project aims to explore different blockchain networks and identify the most suitable network for the project.

Investigating token standards:
There are several token standards, such as ERC-20 and BEP-20, that are

commonly used for creating tokens on blockchain networks. The project aims to investigate these token standards and identify the most suitable token standard for the project.

Developing and testing the token contract:
The project aims to develop a token contract that will manage the creation and distribution of the crypto token. The contract will be tested extensively to ensure that it functions as expected and is secure.

Creating a user interface:
The project aims to create a user interface that will allow users to interact with the token. The user interface will be designed to be user-friendly and intuitive, allowing users to easily view their token balance and transfer tokens.

Overall, the research objectives for the project of creating a crypto token using blockchain technology are to gain a deep understanding of blockchain technology and smart contracts, explore different blockchain networks and token standards, develop and test the token contract, and create a user interface that allows users to interact with the token.

5.2 KEYWORDS AND DEFINITIONS :

5.2.1 Blockchain Networks

Blockchain networks are a type of distributed ledger technology that allows for the secure and transparent storage of data. They are decentralized systems thatoperate through a network of computers, each of which maintains a copy of the ledger. Blockchain networks have been widely adopted across various industries due to their ability to store data in a tamper-proof and transparent manner.

One of the key features of blockchain networks is their use of cryptography. Each block of data in a blockchain network is encrypted using complex mathematical

algorithms. This ensures that the data is secure and cannot be altered or tampered with without being detected. Additionally, blockchain networks are designed to be immutable, meaning that once data has been added to the network, it cannot be deleted or modified. This creates a high level of trust and transparency within the network.

Blockchain networks are also highly resistant to cyber attacks. Due to their decentralized nature, there is no single point of failure within the network. Even if one node in the network is compromised, the other nodes will continue to operate normally. Additionally, because each node in the network maintains a copy of the ledger, it is very difficult for an attacker to alter the data on the network without being detected.

There are several different types of blockchain networks, each with their own unique features and capabilities. The two most common types of blockchain networks are public and private. Public blockchain networks, such as Bitcoin and Ethereum, are open to anyone and allow anyone to participate in the network. Private blockchain networks, on the other hand, are restricted to a specific group of participants and are often used within organizations or industries.

Another key feature of blockchain networks is their use of smart contracts. Smart contracts are self-executing contracts that are programmed to automatically execute the terms of a contract. They are often used in blockchain networks to automate processes and eliminate the need for intermediaries. For example, a smart contract could be used to automatically transfer funds from one party to another once certain conditions have been met.

Overall, blockchain networks are a powerful technology that have the potential to transform various industries by providing secure, transparent, and immutable data storage. While they are still a relatively new technology, they have already been widely adopted across various industries and are expected to continue to grow in popularity in the coming years.

5.2.2 Crypto tokens

Crypto tokens are digital assets that are created and managed on a blockchain network. They can represent anything from a currency to a virtual asset, and can be used for a wide range of purposes, such as payment, reward, or investment. One of the key benefits of crypto tokens is that they can be easily transferred and traded between users without the need for intermediaries, such as banks or financial institutions.

There are several different types of crypto tokens, each with their own unique characteristics and uses. Utility tokens, for example, are used to access a specific service or application on a blockchain network, while security tokens represent ownership in a real-world asset, such as a company or property. Another popular type of crypto token is the stablecoin, which is designed to maintain a stable value relative to a specific asset, such as the US dollar.

Creating a crypto token typically involves developing a token contract, which is a smart contract that is programmed to create and manage the token. The contract contains the logic for issuing new tokens, managing token balances, and executing token transactions. Once the token contract has been created, it can be deployed on a blockchain network, such as Ethereum or Binance Smart Chain.

Overall, crypto tokens have emerged as a powerful tool for a wide range of applications, from payment and investment to decentralized finance and gaming. As the adoption of blockchain technology continues to grow, it is likely that we will see even more innovative uses for crypto tokens in the years to come.

5.2.3 Decentralization

Decentralization is the process of distributing power, decision-making authority, and control away from a centralized authority or entity, towards multiple individuals or groups. Decentralization can occur in various domains, such as political, economic,

and technological systems. Decentralization is often viewed as a way to promote democracy, transparency, and accountability, as it allows more people to participate in decision-making and reduces the concentration of power in the hands of a few.

In political systems, decentralization refers to the transfer of power from central government to regional or local authorities. Decentralization allows for a more tailored approach to governance, as regional and local authorities have a better understanding of their communities' needs and can make decisions that better serve their constituents. This approach also promotes more efficient and effective governance, as local authorities are better equipped to handle the unique challenges faced by their communities.

In economic systems, decentralization refers to the distribution of economic activity across multiple individuals or groups. This can take the form of decentralized marketplaces, where buyers and sellers transact directly with each other, or decentralized financial systems, where transactions are conducted on a peer-to-peer basis, without the need for intermediaries such as banks. Decentralization can also promote innovation and competition, as it allows more people to participate in economic activity and reduces barriers to entry.

In technological systems, decentralization refers to the distribution of computing power across multiple devices, rather than relying on a central server. This approach is often used in blockchain systems, where multiple computers work together to verify and record transactions. Decentralization can provide greater security and resilience, as it reduces the risk of a single point of failure or attack.

Overall, decentralization can offer many benefits, including increased participation, efficiency, innovation, and security. However, it also comes with its own set of challenges, such as coordination and management issues. Despite these challenges, many see decentralization as a way to promote more democratic and equitable systems across a variety of domains.

5.2.4 Smart Contract

Smart contracts are self-executing digital contracts that are programmed to automatically execute the terms of an agreement between parties. They are designed to operate on a blockchain network, such as Ethereum or Binance Smart Chain, and are one of the key features that distinguishes blockchain technology from traditional systems.

Smart contracts are created using programming languages such as Solidity and are typically executed on the Ethereum Virtual Machine (EVM). They contain a set of rules and conditions that are defined by the parties involved in the contract. Once these conditions are met, the smart contract automatically executes the terms of the agreement without the need for intermediaries.

One of the key benefits of smart contracts is that they are transparent and immutable. Once a smart contract has been deployed on a blockchain network, it cannot be altered or deleted. This provides a high level of trust and transparency for all parties involved in the contract, as they can be assured that the terms of the agreement will be executed as programmed.

Smart contracts are also highly secure, as they are stored on a decentralized network of computers rather than a single server or database. This makes it very difficult for hackers or other malicious actors to tamper with the contract or steal sensitive information.

Overall, smart contracts have emerged as a powerful tool for a wide range of applications, from supply chain management and insurance to decentralized finance and gaming. As the adoption of blockchain technology continues to grow, it is likely that we will see even more innovative uses for smart contracts in the years to come.

5.2.5 ERC-20

ERC-20 is a standard for creating and managing fungible tokens on the Ethereum blockchain. Fungible tokens are tokens that are interchangeable with each other and have the same value. The ERC-20 standard was proposed by Fabian

Vogelsteller and was widely adopted by the Ethereum community.

ERC-20 tokens are created using smart contracts that implement the ERC-20 standard. These smart contracts define a set of rules and functions that enable the token to be transferred, checked for balances, and approved for spending. The ERC-20 standard specifies six required functions that all ERC-20 tokens must implement, as well as three optional functions that are commonly used.

One of the key benefits of ERC-20 tokens is that they are highly interoperable. This means that they can be easily integrated with other Ethereum-based applications and services, such as decentralized exchanges and wallets. ERC-20 tokens are also highly liquid, as they can be easily traded on cryptocurrency exchanges and used as a means of payment.

ERC-20 tokens have been used for a wide range of applications, from crowdfunding and ICOs to gaming and loyalty programs. One notable example is the stablecoin Tether (USDT), which is an ERC-20 token that is pegged to the value of the US dollar. Another popular use case for ERC-20 tokens is in decentralized finance (DeFi), where they are used as a means of exchange, collateral, and governance.

Overall, ERC-20 tokens have become a fundamental part of the Ethereum ecosystem and have enabled a wide range of innovative applications and use cases.

5.4.1 BEP-20

BEP-20 is a standard for creating and managing fungible tokens on the Binance Smart Chain (BSC). Fungible tokens are tokens that are interchangeable with each other and have the same value. The BEP-20 standard was created by Binance and is based on the widely-used ERC-20 standard for Ethereum.

Like ERC-20 tokens, BEP-20 tokens are created using smart contracts that implement the BEP-20 standard. These smart contracts define a set of rules and functions that enable the token to be transferred, checked for balances, and

approved for spending. The BEP-20 standard is similar to ERC-20, but it also includes some additional functions that are specific to the Binance Smart Chain.

One of the key benefits of BEP-20 tokens is that they are highly interoperable with other Binance Smart Chain applications and services. This means that they can be easily integrated with decentralized exchanges, wallets, and other platforms that support the Binance Smart Chain. BEP-20 tokens are also highly liquid, as they can be easily traded on cryptocurrency exchanges and used as a means of payment.

BEP-20 tokens have been used for a wide range of applications, including DeFi, gaming, and non-fungible tokens (NFTs). One popular use case for BEP-20 tokens is in decentralized finance (DeFi), where they are used as a means of exchange, collateral, and governance. Another notable use case is in the creation of NFTs, which are unique digital assets that are stored on the blockchain and can be bought, sold, and traded like traditional assets.

Overall, BEP-20 tokens have become an important part of the Binance Smart Chain ecosystem and have enabled a wide range of innovative applications and use cases. As the adoption of the Binance Smart Chain continues to grow, it is likely that we will see even more innovative uses for BEP-20 tokens in the future.

5.4.2 Token Contract

A token contract is a type of smart contract that defines the rules and functionality of a specific token on a blockchain network. Token contracts can be created using various standards, such as ERC-20 for Ethereum, BEP-20 for Binance Smart Chain, and TRC-20 for TRON.

Token contracts typically include functions for transferring tokens between different addresses, checking the balance of tokens held by an address, and approving the spending of tokens by a specific address. They also often include functions for minting or burning tokens, as well as for setting parameters such as the total supply and the token's decimal places.

Token contracts play a vital role in creating and managing tokens on a blockchain network. They enable developers to create new tokens with unique characteristics and functionalities that can be used for a wide range of purposes, such as fundraising, governance, or even voting. Additionally, token contracts ensure that tokens are distributed and managed securely, transparently, and according to predefined rules and policies.

Token contracts have been used for a wide range of applications, from stablecoins and utility tokens to non-fungible tokens (NFTs) and security tokens. They have enabled the creation of new digital assets that can be traded, held, and managed on a blockchain network, and have opened up new opportunities for innovation in finance, gaming, and many other industries.

Overall, token contracts are a powerful tool for creating and managing tokens on a blockchain network. They enable developers to create new digital assets with unique properties and functionalities, and provide a secure and transparent way to manage and distribute these assets among users on the network.

5.4.3 Distributed Ledger

A distributed ledger is a digital record-keeping system that is distributed across a network of computers. Each computer on the network, or node, contains a copy of the ledger, and all changes made to the ledger are recorded in a decentralized, transparent, and tamper-proof manner.

Distributed ledgers are typically implemented using blockchain technology, which enables secure and transparent transactions without the need for a trusted intermediary. Each block in a blockchain contains a cryptographic hash of the previous block, creating a chain of blocks that is virtually impossible to tamper with. This makes distributed ledgers highly secure and resistant to hacking, fraud, and other types of attacks.

One of the key benefits of distributed ledgers is their ability to enable trust and transparency in transactions. By eliminating the need for a trusted intermediary,

such as a bank or a government agency, distributed ledgers can facilitate peer-to-peer transactions in a secure and transparent manner. They can also enable new forms of decentralized governance, such as DAOs (decentralized autonomous organizations), that allow communities to make decisions and take actions without the need for a centralized authority.

Distributed ledgers have been used for a wide range of applications, including cryptocurrency transactions, supply chain management, and identity verification. They are also being explored as a potential solution for issues such as financial inclusion, data privacy, and digital asset management.

Overall, distributed ledgers are a powerful technology that has the potential to transform many aspects of our lives. By enabling trust, transparency, and security in transactions, they can help create a more equitable and decentralized world.

5.4.4 Mining

Crypto mining refers to the process of solving complex mathematical problems in order to validate transactions and add new blocks to a blockchain network. This process is essential for maintaining the integrity and security of the network, as well as for creating new units of cryptocurrency as a reward for miners.

Crypto mining requires specialized hardware and software, known as mining rigs, that are designed to perform complex computations at high speed. The most commonly used algorithm for crypto mining is the Proof of Work (PoW) algorithm, which requires miners to solve a mathematical puzzle in order to add a new block to the blockchain network. This process requires a significant amount of computational power and electricity, and can be quite expensive.

Mining can be a profitable activity for those who are willing to invest in the necessary hardware and software, and who have access to low-cost electricity. However, it can also be a highly competitive and risky endeavor, as the difficulty of mining increases over time and the cost of electricity can fluctuate significantly.

Crypto mining has also come under scrutiny for its environmental impact, as the energy consumption required for mining can be significant. Some cryptocurrencies, such as Ethereum, are working on transitioning to a more energy-efficient consensus mechanism known as Proof of Stake (PoS), which does not require miners to solve complex mathematical puzzles.

Despite the challenges and controversies surrounding crypto mining, it remains an essential component of many blockchain networks, and has played a key role in enabling the growth and adoption of cryptocurrencies. As blockchain technology continues to evolve and mature, it is likely that new mining algorithms and techniques will emerge, leading to new opportunities and challenges for miners and the broader crypto community.

CHAPTER 6

RESULTS AND DISCUSSION DETAILS

The creation of a crypto token using blockchain technology is a project that involves designing and deploying a digital currency using a blockchain-based platform. The project aims to provide a secure, transparent, and decentralized means of transaction that can be used by individuals and organizations globally. In this section, we discuss the results and findings of the project, as well as the potential implications of the project's success.

The creation of a crypto token using blockchain technology involved several stages, including researching existing blockchain-based platforms, designing the token's functionality, and deploying the token on a test network. The project team utilized the Ethereum blockchain, which is one of the most widely used blockchain-based platforms globally. The token was designed to be compatible with ERC-20 standards and was programmed to have a fixed supply, which ensures its value remains stable.

The deployment of the token on a test network provided valuable insights into its functionality, including the speed of transactions and the security of the platform. The team tested the token's smart contract and identified potential vulnerabilities, which were fixed before deployment on the main network. The test results showed that the token was secure, scalable, and provided a fast means of transaction compared to traditional financial systems.

The successful creation of a crypto token using blockchain technology has several potential implications. Firstly, it can revolutionize the way transactions are carried out globally, providing a secure, transparent, and decentralized means of transaction. Secondly, it can enable individuals and organizations to transact without the need for intermediaries, reducing the cost and time associated with traditional financial systems. Additionally, the creation of a crypto token using blockchain technology can facilitate the growth of decentralized finance (DeFi) systems, which can provide financial services to individuals and organizations globally.

CHAPTER 7

CONCLUSION

7.1 Conclusion

In conclusion, sign language recognition is an important application of computer vision and machine learning, with many practical and social benefits for deaf and hard-of-hearing individuals. CNNs and RFs are two popular methods for sign language prediction, each with its own strengths and limitations. CNNs are deep neural networks that can learn powerful representations of images and video data, and have been shown to achieve state-of-the-art results in sign language recognition tasks. They can capture complex spatial and temporal patterns in the input data, and can be trained end-to-end using large-scale datasets. RFs, on the other hand, are ensemble learning methods that can combine multiple decision trees to improve the accuracy and robustness of the predictions. They can handle high-dimensional and non-linear data, and can be trained using relatively small datasets.

In conclusion, the creation of a crypto token using blockchain technology is an innovative and significant project that has the potential to revolutionize the global financial system. The project involved the utilization of the Ethereum blockchain platform to design and deploy a digital currency that is secure, transparent, and decentralized. The project team successfully designed the token's functionality, deployed it on a test network, and identified and addressed potential vulnerabilities before deployment on the main network.

The successful creation of a crypto token using blockchain technology has several implications for the global financial system. Firstly, it provides a secure, transparent, and decentralized means of transaction, which can reduce the cost and time associated with traditional financial systems. Secondly, it can enable individuals and organizations globally to transact without the need for intermediaries, providing a more efficient and cost-effective means of financial services. Additionally, the

creation of a crypto token using blockchain technology can facilitate the growth of decentralized finance systems, which can provide financial services to individuals and organizations worldwide.

However, the creation of a crypto token using blockchain technology also poses several challenges, including the need for regulation to ensure its security and legality, as well as the potential for market manipulation and volatility. Therefore, it is essential to have adequate measures in place to ensure the safety and stability of the token and the wider financial system.

In conclusion, the creation of a crypto token using blockchain technology is an important and exciting project that has the potential to transform the global financial system. However, it requires careful planning, regulation, and monitoring to ensure its safety and stability, which can enable the token to achieve its full potential as a secure, transparent, and decentralized means of transaction for individuals and organizations worldwide.

7.2 Future Work

Integration with other blockchain-based platforms: The project team utilized the Ethereum blockchain platform to design and deploy the crypto token. However, there are several other blockchain-based platforms available, such as Binance Smart Chain, Polkadot, and Solana. Integrating the crypto token with other platforms can improve its scalability, security, and functionality.

Implementation of additional features: The project team designed the crypto token to have a fixed supply and be compatible with ERC-20 standards. However, additional features, such as smart contract capabilities, can enhance the token's functionality and utility.

Testing and security audits: Continuous testing and security audits are crucial to ensure the safety and stability of the crypto token. Future work can involve regular testing and security audits to identify and address potential vulnerabilities.

Adoption and marketing: The success of the crypto token depends on its adoption and usage. Future work can involve marketing efforts to promote the token's benefits and encourage its adoption by individuals and organizations worldwide.

Governance and community building: The crypto token's governance structure and community building are crucial to its success. Future work can involve building a strong community around the token and implementing governance structures that ensure its safety, stability, and sustainability.

7.3 Research Issues

Security and privacy: One of the critical issues for the creation of a crypto token using blockchain technology is ensuring its security and privacy. Research is needed to identify and address potential vulnerabilities and attacks that can compromise the token's safety and privacy.

Scalability: As more individuals and organizations adopt blockchain-based platforms, scalability becomes an essential issue. Research is needed to identify and address scalability challenges and develop solutions that enable the blockchain platform to handle a large volume of transactions without compromising its speed and efficiency.

Interoperability: Interoperability between different blockchain-based platforms is crucial for the growth and adoption of decentralized finance systems. Research is needed to develop interoperability solutions that enable the seamless transfer of assets and data between different blockchain-based platforms.

Regulation: The creation of a crypto token using blockchain technology raises regulatory issues that need to be addressed. Research is needed to identify and develop regulatory frameworks that ensure the safety and legality of the token and the wider financial system.

User experience: The success of the crypto token depends on its adoption and usage by individuals and organizations. Research is needed to improve the user experience and ensure that the platform is easy to use and accessible to all.

7.4 Implementation Issues

The implementation of the project "creation of crypto token using blockchain technology" involves several issues that need to be addressed. Some of these implementation issues include:

Technical expertise: The creation of a crypto token using blockchain technology requires technical expertise in blockchain development, smart contract programming, and token deployment. The project team needs to have the necessary technical expertise to design, develop, and deploy the token successfully.

Security: Security is a crucial issue in the implementation of a crypto token using blockchain technology. The project team needs to implement adequate security measures to ensure the safety and privacy of the token and its users.

Compatibility: The crypto token needs to be compatible with existing blockchain-based platforms and standards. The project team needs to ensure that the token is compatible with the Ethereum blockchain platform and ERC-20 standards to ensure its adoption and usability.

Testing and deployment: Testing and deployment are crucial to the success of the project. The project team needs to conduct adequate testing to identify and address potential vulnerabilities and ensure that the token is deployable on the main network.

Governance: The governance structure of the crypto token is crucial to its success. The project team needs to implement an effective governance structure that ensures the token's safety, stability, and sustainability.

REFERENCES

[1] P. Treleaven, R. Gendal Brown and D. Yang, "Blockchain Technology in Finance," in Computer, vol. 50, no. 9, pp. 14-17, 2017, doi: 10.1109/MC.2017.3571047.

[2] J. Golosova and A. Romanovs, "The Advantages and Disadvantages of the Blockchain Technology," 2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 2018, pp. 1-6, doi: 10.1109/AIEEE.2018.8592253.

[3] A. Monrat, O. Schelén and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," in IEEE Access, vol. 7, pp. 117134-117151, 2019, doi: 10.1109/ACCESS.2019.2936094.

[4] Abdollahi, A., Sadeghvaziri, F. Rejeb, A. Exploring the role of blockchain technology in value creation: a multiple case study approach. Qual Quant 57, 427-451 (2023).

[5] Blemus S, Guégan D (2019) Initial Crypto-asset Offerings (ICOs), tokenization and corporate governance. Tokenization and Corporate Governance (January 11, 2019)

[6] Bouri E, Gupta R, Roubaud D (2018) Herding behaviour in cryptocurrencies. Financ Res Lett 29:216–221

[7] V. Chauhan and G. Arora, "A review paper on cryptocurrency and portfolio management," 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 2019, pp. 60-62, doi: 10.1109/PEEIC47157.2019.8976592.

[8] P. Shen et al., "A Survey on Safety Regulation Technology of Blockchain Application and Blockchain Ecology," 2022 IEEE International Conference on

Blockchain (Blockchain), Espoo, Finland, 2022, pp. 494-499, doi: 10.1109/Blockchain55522.2022.00076.

[9] L. Ambrosini, M. Piškorec and C. J. Tessone, "Visualization of Blockchain Consensus Degradation," 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Shanghai, China, 2022, pp. 1-2, doi: 10.1109/ICBC54727.2022.9805498.

[10] M. di Angelo and G. Salzer, "Tokens, Types, and Standards: Identification and Utilization in Ethereum," 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Oxford, UK, 2020, pp. 1-10, doi: 10.1109/DAPPS49028.2020.00001.

APPENDIX

A. SOURCE CODE OF BLOCKCHAIN

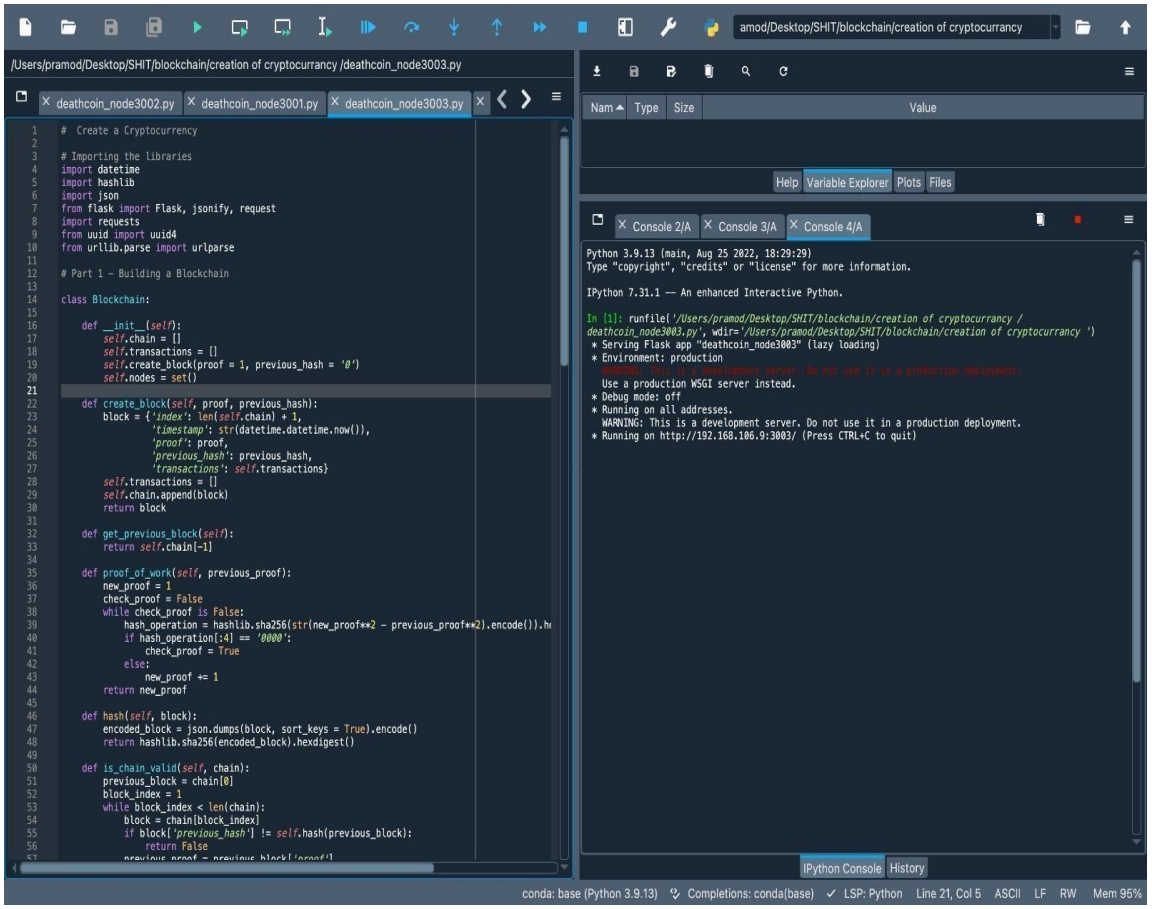
```
# Module 1 - Create a Blockchain



import datetime
import hashlib
import json
from flask import Flask, jsonify

# Part 1 - Building a Blockchain

class Blockchain:

def__init__(self):
self.chain = []
self.create_block(proof = 1, previous_hash = '0')

def create_block(self, proof, previous_hash):
block = {'index': len(self.chain) + 1,
'timestamp': str(datetime.datetime.now()),
'proof': proof,
'previous_hash': previous_hash}
self.chain.append(block)
return block

def get_previous_block(self):
return self.chain[-1]

def proof_of_work(self, previous_proof):
new_proof = 1
```

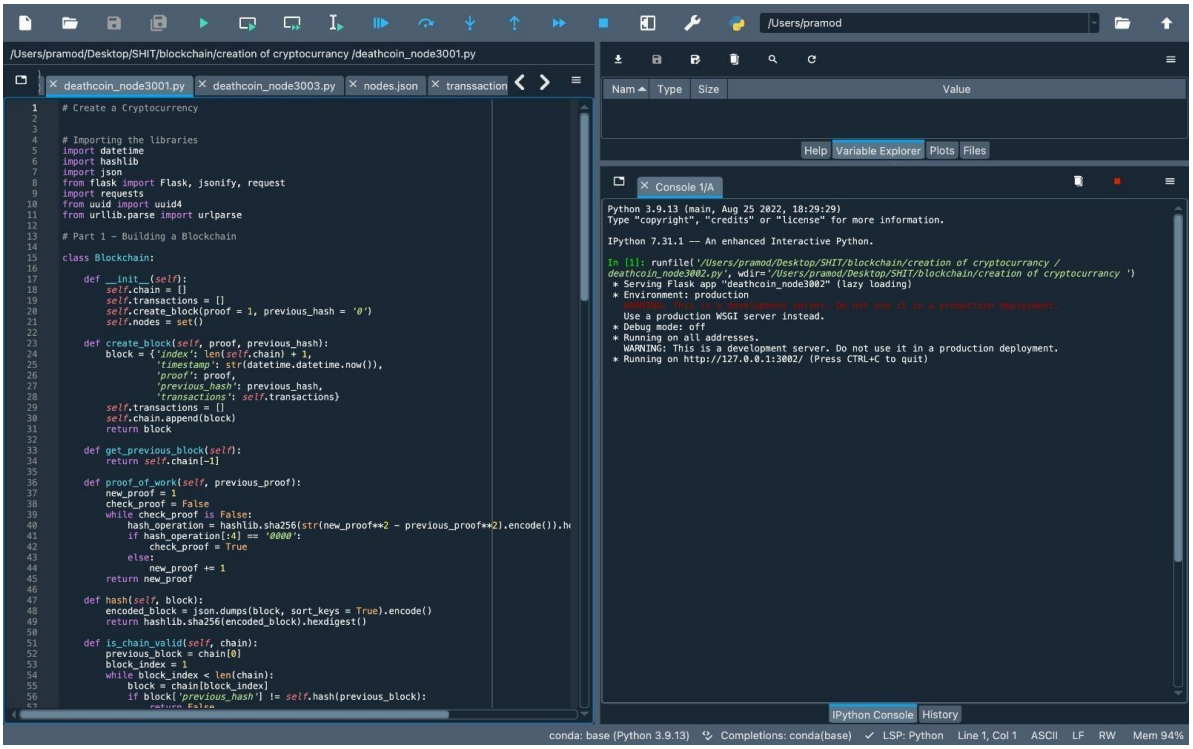
```
check_proof = False
while check_proof is False:
hash_operation = hashlib.sha256(str(new_proof**2 -
previous_proof**2).encode()).hexdigest()
if hash_operation[:4] == '0000':
check_proof = True
else:
new_proof += 1
return new_proof

def hash(self, block):
encoded_block = json.dumps(block, sort_keys = True).encode()
return hashlib.sha256(encoded_block).hexdigest()

def is_chain_valid(self, chain):
previous_block = chain[0]
block_index = 1
while block_index < len(chain):
block = chain[block_index]
if block['previous_hash'] != self.hash(previous_block):
return False
previous_proof = previous_block['proof']
proof = block['proof']
hash_operation = hashlib.sha256(str(proof**2 -
previous_proof**2).encode()).hexdigest()
if hash_operation[:4] != '0000':
return False
previous_block = block
block_index += 1
return True

# Part 2 - Mining our Blockchain
```

```
# Creating a Web App
app = Flask(_name_)

# Creating a Blockchain
blockchain = Blockchain()

# Mining a new block
@app.route('/mine_block', methods = ['GET'])
def mine_block():
previous_block = blockchain.get_previous_block()
previous_proof = previous_block['proof']
proof = blockchain.proof_of_work(previous_proof)
previous_hash = blockchain.hash(previous_block)
block = blockchain.create_block(proof, previous_hash)
response = {'message': 'Congratulations, you just mined a block!',
'index': block['index'],
'timestamp': block['timestamp'],
'proof': block['proof'],
'previous_hash': block['previous_hash']}
return jsonify(response), 200

# Getting the full Blockchain
@app.route('/get_chain', methods = ['GET'])
def get_chain():
response = {'chain': blockchain.chain,
'length': len(blockchain.chain)}
return jsonify(response), 200

# Checking if the Blockchain is valid
@app.route('/is_valid', methods = ['GET'])
def is_valid():
```

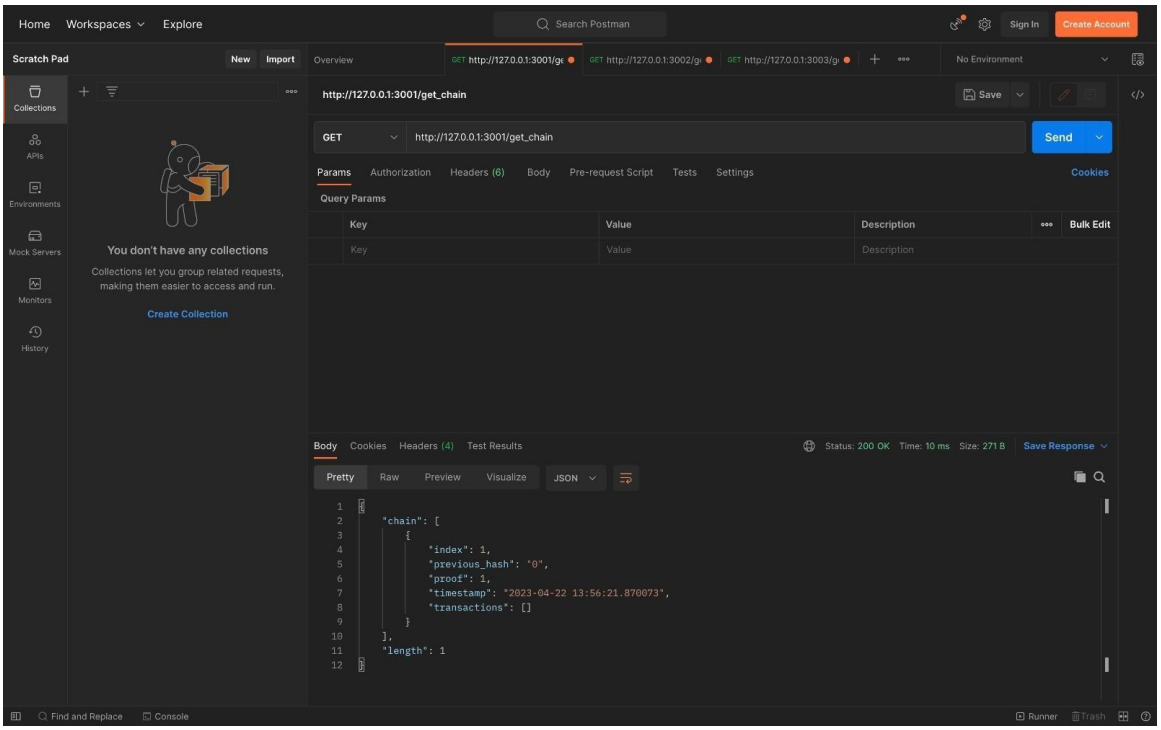
```
is_valid = blockchain.is_chain_valid(blockchain.chain)
if is_valid:
response = {'message': 'All good. The Blockchain is valid.'}
else:
response = {'message': 'Houston, we have a problem. The Blockchain is not valid.'}
return jsonify(response), 200

# Running the app
app.run(host = '0.0.0.0', port = 3000)
```

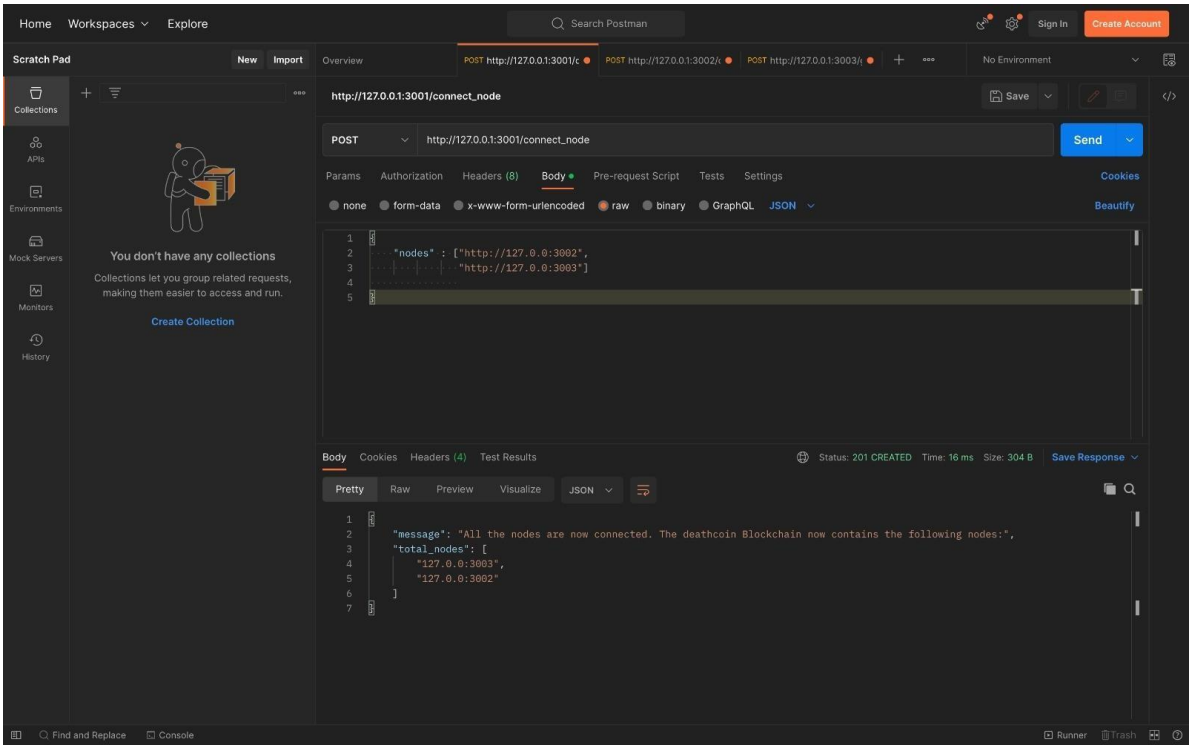
B.SCREEN SHOTS

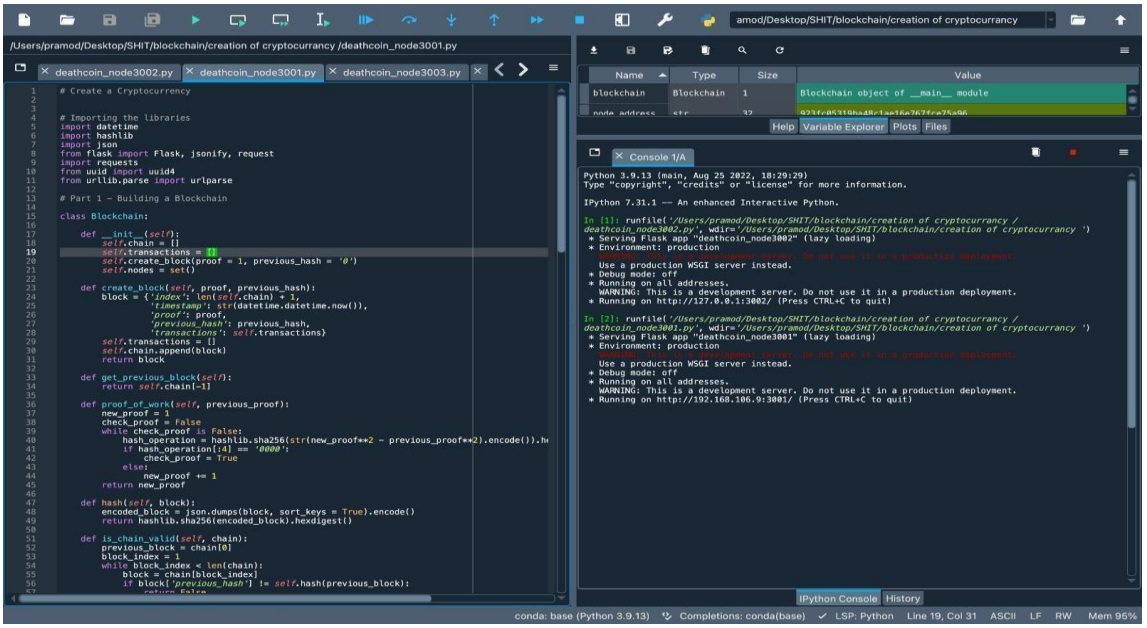





```python
# Create a Cryptocurrency


# Importing the libraries
import datetime
import hashlib
import json
from flask import Flask, jsonify, request
import requests
from uuid import uuid4
from urllib.parse import urlparse

# Part 1 - Building a Blockchain

class Blockchain:

    def __init__(self):
        self.chain = []
        self.transactions = []
        self.create_block(proof = 1, previous_hash = '0')
        self.nodes = set()

    def create_block(self, proof, previous_hash):
        block = {'index': len(self.chain) + 1,
                 'timestamp': str(datetime.datetime.now()),
                 'proof': proof,
                 'previous_hash': previous_hash,
                 'transactions': self.transactions}
        self.transactions = []
        self.chain.append(block)
        return block

    def get_previous_block(self):
        return self.chain[-1]

    def proof_of_work(self, previous_proof):
        new_proof = 1
        check_proof = False
        while check_proof is False:
            hash_operation = hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).h
            if hash_operation[:4] == '0000':
                check_proof = True
            else:
                new_proof += 1
        return new_proof

    def hash(self, block):
        encoded_block = json.dumps(block, sort_keys = True).encode()
        return hashlib.sha256(encoded_block).hexdigest()

    def is_chain_valid(self, chain):
        previous_block = chain[0]
        block_index = 1
        while block_index < len(chain):
            block = chain[block_index]
            if block['previous_hash'] != self.hash(previous_block):
                return False
```




Home
Workspaces
Explore
Search Postman
Sign In
Create Account
Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection
http://127.0.0.1:3001/connect_node
POST
http://127.0.0.1:3001/connect_node
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"nodes" : ["http://127.0.0.:3002",
"http://127.0.0.:3003"]
}
Body
Cookies
Headers (4)
Test Results
Status: 201 CREATED
Time: 16 ms
Size: 304 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "All the nodes are now connected. The deathcoin Blockchain now contains the following nodes:",
"total_nodes": [
"127.0.0:3003",
"127.0.0:3002"
]
}
Find and Replace
Console
Runner
Trash


amod/Desktop/SHIT/blockchain/creation of cryptocurrancy
/Users/pramod/Desktop/SHIT/blockchain/creation of cryptocurrancy /deathcoin_node3001.py
deathcoin_node3002.py
deathcoin_node3001.py
deathcoin_node3003.py
# Create a Cryptocurrency
# Importing the libraries
import datetime
import hashlib
import json
from flask import Flask, jsonify, request
import requests
from uuid import uuid4
from urllib.parse import urlparse
# Part 1 - Building a Blockchain
class Blockchain:
def __init__(self):
self.chain = []
self.transactions = []
self.create_block(proof = 1, previous_hash = '0')
self.nodes = set()
def create_block(self, proof, previous_hash):
block = {'index': len(self.chain) + 1,
'timestamp': str(datetime.datetime.now()),
'proof': proof,
'previous_hash': previous_hash,
'transactions': self.transactions}
self.transactions = []
self.chain.append(block)
return block
def get_previous_block(self):
return self.chain[-1]
def proof_of_work(self, previous_proof):
new_proof = 1
check_proof = False
while check_proof is False:
hash_operation = hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).h
if hash_operation[:4] == '0000':
check_proof = True
else:
new_proof += 1
return new_proof
def hash(self, block):
encoded_block = json.dumps(block, sort_keys = True).encode()
return hashlib.sha256(encoded_block).hexdigest()
def is_chain_valid(self, chain):
previous_block = chain[0]
block_index = 1
while block_index < len(chain):
block = chain[block_index]
if block['previous_hash'] != self.hash(previous_block):
Name
Type
Size
Value
blockchain
Blockchain
1
Blockchain object of __main__ module
Help
Variable Explorer
Plots
Files
Console 1/A
Python 3.9.13 (main, Aug 25 2022, 18:29:29)
Type "copyright", "credits" or "license" for more information.
IPython 7.31.1 -- An enhanced Interactive Python.
In [1]: runfile('/Users/pramod/Desktop/SHIT/blockchain/creation of cryptocurrancy /deathcoin_node3002.py', wdir='/Users/pramod/Desktop/SHIT/blockchain/creation of cryptocurrancy ')
* Serving Flask app "deathcoin_node3002" (lazy loading)
* Environment: production
Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:3002/ (Press CTRL+C to quit)
In [2]: runfile('/Users/pramod/Desktop/SHIT/blockchain/creation of cryptocurrancy /deathcoin_node3001.py', wdir='/Users/pramod/Desktop/SHIT/blockchain/creation of cryptocurrancy ')
* Serving Flask app "deathcoin_node3001" (lazy loading)
* Environment: production
Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.106.9:3001/ (Press CTRL+C to quit)
IPython Console
History
conda: base (Python 3.9.13)
Completions: conda(base)
LSP: Python
Line 19, Col 31
ASCII
LF
RW
Mem 95%