

SMART TENDER CONTRACT MANAGEMENT SYSTEM IN PYTHON USING BLOCKCHAIN

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology in Information Technology

By

SRINIVASA RAO (Reg.No – 40120063)
VENKATA BHASKAR (Reg.No – 40120126)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Category - 1 University by UGC

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

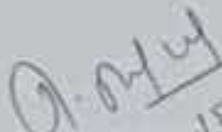
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

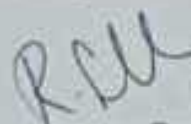
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Srinivasa Rao (40120063)** and **Venkata Bhaskar (40120126)** who carried out the Project entitled **"SMART TENDER CONTRACT MANAGEMENT SYSTEM IN PYTHON USING BLOCKCHAIN"** under my supervision from January 2024 to April 2024.


15/4/24
Internal Guide

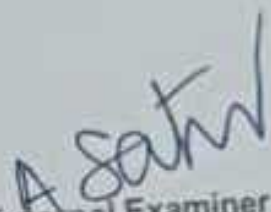
Ms. S. Sageengrana M.E.,


15/4/24
Head of the Department

Dr. R. Subhashini, M.E., Ph.D.,

Submitted for Viva voce Examination held on 12/04/24


Internal Examiner


External Examiner

DECLARATION

I, **Srinivasa Rao (40120063)**, hereby declare that the Project Report entitled "**Smart Tender Contract Management System In Python Using Blockchain**" done by me under the guidance of **Ms. S. Sageengrana M.E.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 17/4/24

PLACE: Chennai

N. Srinilasa Rao

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. R. Subhashini M.E., Ph.D.**, Head of the Department of Information Technology 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.S.Sageengrana M.E.**, for her valuable guidance, suggestions and constant encouragement paved 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 Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

The offering interaction is a key business measure that assists organizations with tracking down an appropriate worker for hire. In the structure and development enterprises, customers welcome givers to present a gauge of costs, enumerating the expenses related with finishing a structure. Thusly, the customer can put together their choice with respect to the consequence of the tenders to choose the most reasonable project worker. The givers need to gather the delicate's booklet, value it, and take it back to the customer's office before the cut-off time. In this paper, we present a plan and execution of an e-offering framework by utilizing Web administrations for the mechanization of such offering measures. In this paper, we utilized the blockchain innovation in advanced e-offering framework to tackle the security issues and satisfy the framework necessities.

The system employs smart contracts, self-executing agreements with the terms of the contract directly written into code, eliminating the need for intermediaries and ensuring trust and transparency among parties involved. Through blockchain integration, STCMS ensures data integrity, traceability, and tamper-resistance, mitigating the risks of fraud and manipulation commonly associated with traditional tender management systems.

Key features of STCMS include automated contract execution based on predefined conditions, real-time monitoring of contract status, and seamless integration with existing tender management workflows. Additionally, the system incorporates advanced cryptographic techniques to secure sensitive data and ensure privacy while maintaining accessibility and auditability.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	iv
	LIST OF FIGURES	vii
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
	2.1 Inferences from Literature Survey	
	2.2 Open problems in Existing System	5
3	REQUIREMENTS ANALYSIS	6
	3.1 Feasibility Studies/Risk Analysis of the Project	6
	3.2 Software Requirements Specification Document	7
4	DISCRIPTION OF PROPOSED SYSTEMS	8
	4.1 Proposed Systems	8
	4.2 Architecture Diagram	9
	4.3 Description of Software for Implementation and Testing Plan	10
5	METHODOLOGY	13
	5.1 Modules	13
	5.2 Data Flow Diagram	14
	5.3 UML Diagrams	15

	5.3.1 Use Case Diagram	16
	5.3.2 Sequence Diagram	17
	5.4 Input Design	18
	5.5 Output Design	19
6	RESULT AND FUTURE SCOPE	21
	6.1 Result	21
	6.2 Future Scope	21
7	CONCLUSION	23
	REFERENCES	24
	APPENDICES	26
	A Source Code	26
	B Screen Shots	41
	C Research Paper	44

LIST OF FIGURES

Figure No	Figure Name	Page No
4.1	System Architecture	9
5.1	Data Flow Diagram	15
5.2	Use Case Diagram	17
5.3	Sequence Diagram	18
B.1	Home Page	41
B.2	Admin Login	41
B.3	Apply for Tender	42
B.4	Admin Dashboard	42
B.5	Tender Status Details	43
B.6	Tender Action Page	43

CHAPTER 1

INTRODUCTION

Current E-Tendering systems are not 'fair and open' meaning that the information is not shared with all stakeholders (Right to Information). The information is released on 'as they please' basis for example - when a company is selected as a winner of a contract, other companies that bid on the same tender are not notified of why their bid was rejected and why a particular company was selected as a winner. A company can request this information but it is a tedious process of getting this data. Even though auditing these documents is possible, evaluating the documents needs time. Aside from not being straightforward, security is additionally a significant issue for these entryways prompting misrepresentation and control of information put away in a brought together data set. If the hacker gets hold of this centralized database, bids can be leaked to competitors leading to major financial and strategic losses for a business. Blockchain innovation can be utilized to settle these security suggestions as it vigorously around the decentralization of data and is gotten by encryption coordinated with irrefutable square based engineering for exchange the executives. Hence, Blockchain can be used as a transparent, decentralized and secured tendering framework that will facilitate bidders' oversight on portal functions and observe all the activities carried out by the tender portal.

CHAPTER 2

LITERATURE REVIEW

2.1 Inferences from literature survey

Wang, Wenbo, et al. [1] “In blockchain networks, a survey on consensus methods and mining strategy management has been conducted.” IEEE Access 7 (2019): 22328-22370.

The fast advancement of blockchain technologies over the last decade has piqued the curiosity of both academics and businesses alike. The blockchain network started as a decentralised, immutable ledger system for transactional data ordering in the Internet banking sector. It is now thought to be a powerful backbone/framework for decentralised data processing and data-driven self-organization in flat, open-access networks. The unique decentralised consensus processes introduced by public blockchain are mostly to blame for the conceivable qualities of decentralization, immutability, and self-organization. The lack of a comprehensive literature study on the development of decentralized consensus mechanisms in blockchain networks inspired this survey. We present a systematic view of blockchain network organization in this study. Our in-depth examination of the state-of-the-art consensus protocols focuses on both the perspective of distributed consensus system design and the perspective of incentive mechanism design, emphasising the unique characteristics of decentralized consensus in blockchain networks. We also give a game-theoretic solution thorough an examination of the approach used by individual nodes in blockchain backbone networks to self-organize. As a result, we present a thorough examination of the growing uses of blockchain networks in a wide range of telecommunications. We are particularly interested in the impact of consensus procedures on these applications. Finally, we go over some of the open difficulties in blockchain consensus protocol design, as well as some of the possible research avenues.

Ambegaonker, Ajeenkya, Utkarsh Gautam, and Radha Krishna Rambola. [2] “Successful Tender process Approach using Block chain technology to Maintain Safety and Integrity.” 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.

The problem with current tendering is that it only reaches a limited number of people; however, the internet is expanding, and tendering is not far behind. We have some online tendering systems, but they are not as secure as they should be because tendering contains confidential data that should not be leaked, and Blockchain solves that problem effectively. The goal of this study is to uncover better ways to tender, because tendering is such an important element of business and development that improving this system would lead to greater development. Time efficiency, employment, and a fair system are some of the elements that the suggested system of this research can enhance.

Zheng, Zibin et al. [3]“An introduction to blockchain technology, including its architecture, consensus, and future developments.”2017 IEEE international congress on big data (Big Data congress). IEEE, 2017.

The blockchain, which is at the heart of Bitcoin, has recently gotten a lot of attention. Blockchain acts as an immutable record that allows decentralised transactions to take place. Blockchain-based applications are gaining traction in a variety of industries, including financial services, reputation management, and the Internet of Things (IoT), among others. However, there are still a number of issues with blockchain technology to address, such as scalability and security concerns. This paper provides an in-depth look of blockchain technology. First, we give an introduction of blockchain architecture before comparing some common consensus methods utilised in various blockchains. There are also brief mentions on technical issues and current improvements. We also discuss likely blockchain developments in the future.

Cachin, Christian, and Marko Vukolić. [4] “Blockchain consensus protocols in the wild”. arXiv:1707.01873 arXiv preprint (2017).

A blockchain is a decentralised ledger for recording transactions that is maintained by numerous nodes without the need of a central authority and is based on a distributed cryptographic system. A consensus mechanism guarantees that all nodes agree on a unique sequence in which entries are attached to the blockchain, and all nodes check the information to be appended to it. Because they also deal with blockchain systems, consensus techniques for tolerating Byzantine flaws have gotten a lot of interest. The process of evaluating and establishing confidence in the robustness of consensus protocols exposed to failures and hostile nodes is discussed in this paper. We propose using standard cryptography and computer security practises such as public reviews, detailed models, and formal proofs; yet, many real-world system designers appear to be unaware of this. We also look at the failure models and attack resilience of the consensus algorithms employed by a variety of well-known permissioned blockchain systems.

Pilkington, Marc. [5]“Principles and uses of blockchain technology.” Research handbook on digital transformations. Edward Elgar Publishing, 2016.

This article explains the fundamentals of blockchain technology as well as some of its most cutting-edge uses. To begin, we'll go over the fundamental principles behind the blockchain, as well as the possible dangers and downsides of public distributed ledgers, as well as the trend toward hybrid solutions. Second, we highlight the key characteristics of decentralised public ledger platforms. We illustrate why blockchain is a revolutionary and foundational technology in the third section, and we sketch out a list of key applications in the fourth section, taking into account recent developments.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Blockchain has become an unavoidable future in enterprise finance, particularly enabling and securing cross-company transactions. By introducing a comparable notion of smart contract, the trusted sub-ledger operation (TSLO), this article will propose a complete architecture based on the Blockchain to solve the traceability and validity of accounting data by assets groupement. TSLO is a more flexible and adaptable method for asset management in the corporate accounting system and the enterprise resource planner. This method is built on a decentralized microservices tree (DMST) and is an extendable E-Bidding form of TEA (Triple Entry Accounting). Instead of using a multi-ledger architecture, the Hyperledger Fabric skeleton, limited to participant channels inside one entity or organization, our approach uses decentralized sub-ledgers with an implementation tree (DMST) for an assets-driven transactions. Furthermore, the government's audit and taxation procedures for financial groups are more accessible by combining Proof of Authority and Proof of Stake to assure the logic of More stake more reputation to preserve.

DIS-ADVANTAGES

It's nearly impossible to change smart financial processes, and every programming defect becomes time-consuming and costly to fix.

Even Though financial contain terms that are not always understood, smart financial are not always capable of dealing with vague terms and conditions.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system is well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

3.2 SOFTWARE REQUIREMENT SPECIFICATION

HARDWARE REQUIREMENTS

System : Pentium i3 Processor
Hard Disk : 500 GB.
Monitor : 15" LED
Input Devices : Keyboard
Mouse Ram : 2 GB

SOFTWARE REQUIREMENTS

Operating system : Windows 10
Coding Language : Python

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEMS

4.1 Proposed Systems

Nowadays, organizations such as agriculture, healthcare, financial services, and even the government sector have started to implement blockchain. Even so, the tendering sector is incompetent to benefit from this new blockchain technology. If tender management is done wrongfully it may lead to huge losses in case of faulty practices. This may include favouring contractors, improper record maintenance, lack of transparency, hacking, data modification, and other similar issues.

To resolve tender management issues, here we use blockchain technology to make sure that it helps in maintaining security and efficiency. With encryption coupled with indisputable block-based architecture, the transaction management blockchain is fully secured.

Here, we make use of blockchain technology to secure transaction-based documents along with transactions such as tender documents, applications, bid proposals, company profiles, past records, approving officer details, and rejection details to ensure a completely transparent tendering process.

ADVANTAGES

The Smart Tender Management System lowers the likelihood of third-party manipulation. Furthermore, the absence of a mediator in smart tenders saves money.

It is encrypted, and cryptography protects all papers from unauthorized access.

It automates the system using computer protocols, saving hours of time in several business processes.

4.2 ARCHITECTURE DIAGRAM

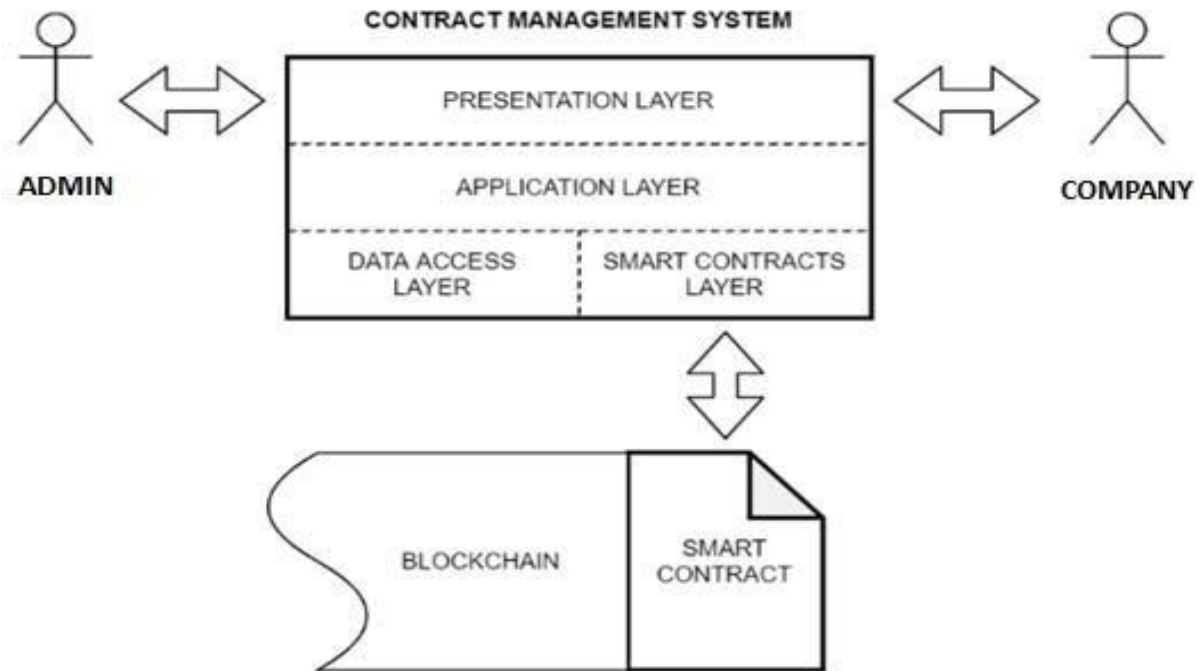


Figure 4.1 System Architecture

4.3 Description of Software for Implementation and Testing plan

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration

testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be

accepted. Invalid Input : identified classes of invalid input must be

rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be

exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a blackbox level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

CHAPTER 5

METHODOLOGY

5.1 MODULES

- Admin Module
- Company Module

MODULES DESCRIPTION:

Admin:

- Login
- Manage Tender
- List of all the tender
- View / update / delete / Add
- Add Tender
- Update Tender
 - Only a few fields which are not very important and likely to have changed can be updated.
- Ongoing Tender
 - list of all the tenders which are open for bids/proposals
 - list of all the applications received for the tender
 - Admin can update the status of the Application (Accept/Reject/Shortlist)
- Past Tenders
 - List of all the tenders which are closed for bids and like ongoing canview applications & update the status
- View Transactions
 - List of tenders & their applications with Manipulation status

- It will indicate if there were any manipulations done without the consent or knowledge of the Admin.
- We will explain the manipulation part at the end of the video.

Company:

- Register
- Login
- Company Profile
- Tenders
 - List of all the ongoing tenders which are open for bids
 - Company can upload their proposal and participate.
- Change Password
- Let's go to the admin panel and view the application and change the status
- Let's go back to the Companies panel
- My Application
- List of my application & Status

5.2. DATA FLOW DIAGRAMS

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to

output. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

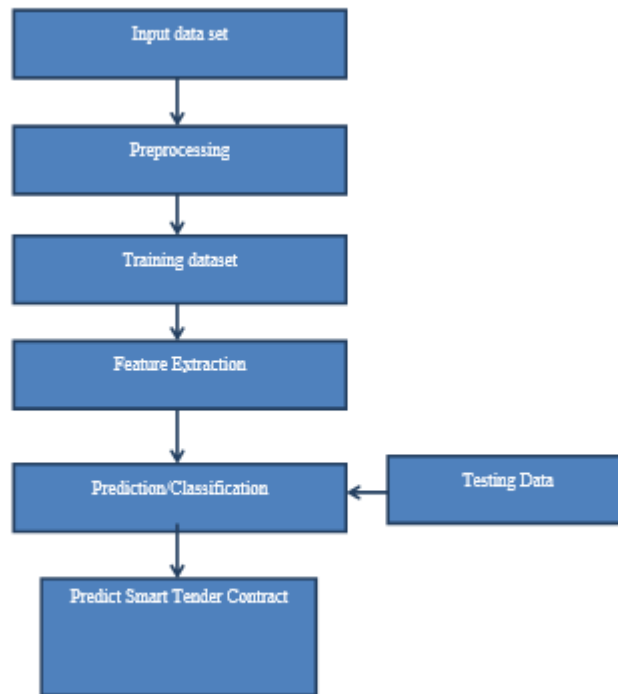


Figure: 5.1 Data Flow Diagram

5.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization,

Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

5.3.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

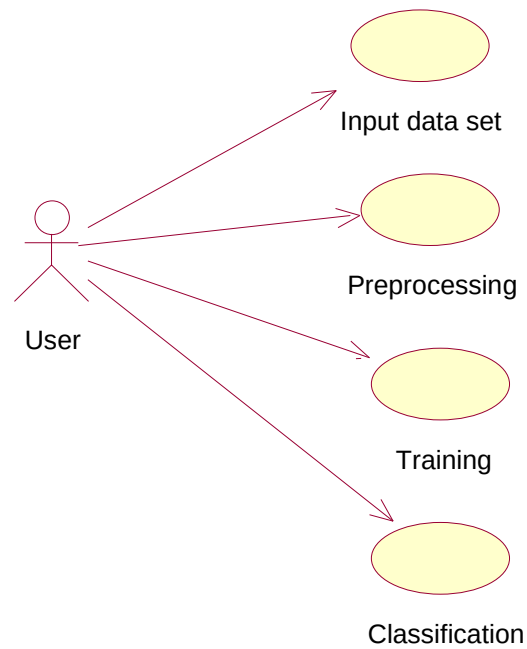


Figure 5.2. Use Case Diagram

5.3.2 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

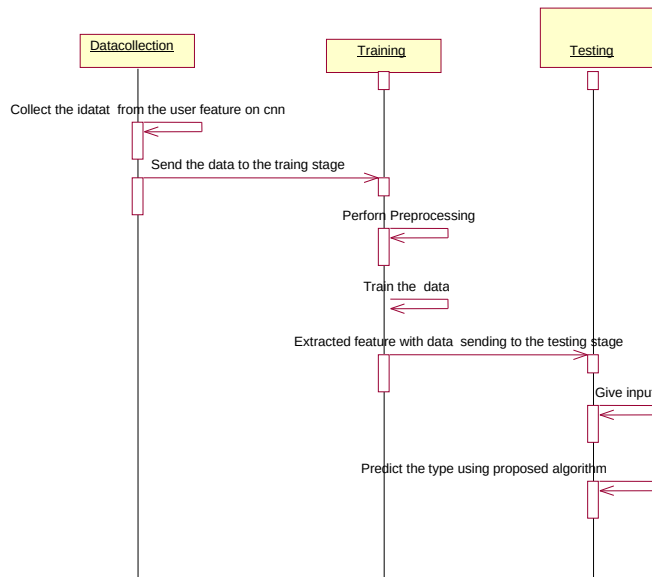


Figure 5.3 Sequence Diagram

5.4 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.

Objectives:

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

5.5 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.

- Create document, report, or other formats that contain information produced by the system.
- The output form of an information system should accomplish one or more of the following objectives.
- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER 6

RESULT AND DISCUSSION

6.1 Result

We use technology based totally transaction documents to secure smooth files, bids, bids, team info, past records, approval information and rejection details. A smart tender contract management system in Python utilizing blockchain technology offers a robust solution for enhancing transparency, security, and efficiency in tender processes. By leveraging blockchain's decentralized ledger, the system ensures tamper-proof record-keeping, eliminating the risk of data manipulation or fraud. Through smart contracts coded in Python, the system automates various aspects of the tender process, such as bid submission, evaluation, and awarding contracts, streamlining operations and reducing administrative overhead. Smart contracts execute predefined rules autonomously, facilitating trustless interactions between parties involved in the tender process. Additionally, the use of blockchain enables real-time tracking and auditing of tender-related transactions, providing stakeholders with a clear and immutable audit trail. This not only enhances accountability but also fosters greater confidence among participants in the integrity of the tender process. Overall, a smart tender contract management system built on Python and blockchain technology revolutionizes traditional tendering methods by promoting transparency, efficiency, and trust in procurement processes.

6.2 Future scope

Looking ahead, the smart tender contract management system developed in Python using blockchain technology holds immense potential for further advancements and future scope. One promising area for future exploration involves expanding the system's capabilities to support a broader range of procurement processes beyond tender contract management. This could include functionalities for supplier management, contract lifecycle management, and procurement analytics, allowing organizations to streamline their entire procurement workflow on a single platform.

Additionally, there is an opportunity to enhance the system's scalability and interoperability to accommodate larger transaction volumes and integrate seamlessly with other blockchain networks and external systems. Furthermore, future iterations of the system could incorporate advanced features such as AI-driven decision support, predictive analytics, and autonomous contract execution, enabling organizations to make data-driven decisions and automate routine procurement tasks. Moreover, there is a growing need for blockchain-based solutions that address sustainability and ethical sourcing concerns in procurement, presenting an exciting avenue for future development. By integrating sustainability metrics, supply chain transparency, and ethical sourcing criteria into the system, organizations can ensure responsible and socially conscious procurement practices. Overall, the future scope for the smart tender contract management system in Python using blockchain technology is vast, encompassing advancements in functionality, scalability, interoperability, sustainability, and automation to meet the evolving needs of modern procurement practices

CHAPTER 7

CONCLUSION

When it involves packages like tender control, security and transparency are vital, traditional technology and information planning strategies can no longer be used as they pose the risk of records corruption. As we realize, delicate systems using centralized databases to make contracts and make orders have many security requirements. SHA and AES algorithms for encryption and the usage of NLP principles to assess patron remarks enables the smooth enterprise to choose the great proposition within the listing. The use of generation including block chain, decentralized databases, can provide the safety and openness vital for these sorts of programs. This article discusses how any such machine can be designed, deliberating the diverse approaches involved, and their underlying implementation.

In the realm of tender control, security and transparency are paramount considerations. Traditional technology and information planning strategies are no longer adequate as they pose significant risks of data corruption and breaches. Centralized databases, commonly used in conventional systems for contract management and order processing, come with numerous security vulnerabilities. To address these concerns, robust encryption techniques such as SHA and AES algorithms are employed to safeguard sensitive data. Additionally, Natural Language Processing (NLP) principles are utilized to analyze customer feedback, aiding in the selection of the most suitable proposals from the list.

Integrating cutting-edge technologies like blockchain and decentralized databases is essential to provide the necessary security and transparency for such critical applications. Blockchain technology ensures immutability and transparency by creating an indelible ledger of transactions distributed across a network of nodes. This decentralized approach not only enhances data integrity but also mitigates the risk of single points of failure and malicious attacks.

REFERENCES:

1. "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967. K. C. Davis, "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967.
2. The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967, "The Information Act: A Preliminary Analysis." "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967. K. C. Davis, "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967.3. Pal, Om, and Surendra Singh.
3. Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin 89-102 in Journal of Information Technology in Construction, vol. 11 (2006).
4. Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE international congress on big data (BigData congress). IEEE, 2017.
5. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.
6. Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media.
7. Tapscott, D., & Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World. Penguin.
8. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE Access, 4, 2292-2303.
9. Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In IEEE International Congress on Big Data (pp. 557-564). IEEE.
10. Swan, M. (2015). Blockchain: Blueprint for a New Economy. O'Reilly Media.
11. Lewis, G. A., Mavridou, T., & Gorbach, A. (2018). A Blockchain-Based Smart Contract System for Healthcare Management. In IEEE International Conference on Software Architecture (ICSA) Workshops (pp. 230-233). IEEE.
12. Zheng, Z., Xie, S., Dai, H. N., & Wang, H. (2017). Blockchain Challenges and

Opportunities: A Survey. *International Journal of Web and Grid Services*, 14(4), 352-375.

13. Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, 14(4), 352-375.
14. Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaria, V. (2018). Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?. *Future Internet*, 10(2), 20.

APPENDICES

A. SOURCE CODE

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ChangeListManager">
    <list default="true" id="8ef498ec-2780-451c-9aef-a06537692f56" name="Changes"
comment="" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="FileTemplateManagerImpl">
    <option name="RECENT_TEMPLATES">
      <list>
        <option value="HTML File" />
      </list>
    </option>
  </component>
  <component name="ProjectId" id="28bRDpkQHcEBRsPUruVRqVRlhg7" />
  <component name="ProjectViewState">
    <option name="hideEmptyMiddlePackages" value="true" />
    <option name="showLibraryContents" value="true" />
  </component>
  <component name="PropertiesComponent">
    <property name="DefaultHtmlFileTemplate" value="HTML File" />
    <property name="RunOnceActivity.OpenProjectViewOnStart" value="true" />
    <property name="RunOnceActivity.ShowReadmeOnStart" value="true" />
    <property name="last_opened_file_path" />
  </component>
</project>
```

```

value="$PROJECT_DIR$/../BloodBankDjangoBhuwan/Blood_Bank" />
</component>
<component name="RecentsManager">
  <key name="CopyFile.RECENT_KEYS">
    <recent name="D:\Project\Python
Project\OnlineFire_ReportingSystem\firereport\templates" />
    <recent name="D:\Project\Python
Project\OnlineFire_ReportingSystem\firereport\static" />
    <recent name="D:\Project\Python
Project\OnlineFire_ReportingSystem\firereport\static\images" />
  </key>
  <key name="MoveFile.RECENT_KEYS">
    <recent name="D:\Project\Python
Project\OnlineFire_ReportingSystem\firereport\templates" />
  </key>
</component>
<component name="SpellCheckerSettings" RuntimeDictionaries="0" Folders="0"
CustomDictionaries="0" DefaultDictionary="application-level"
UseSingleDictionary="true" transferred="true" />
<component name="TaskManager">
  <task active="true" id="Default" summary="Default task">
    <changelist id="8ef498ec-2780-451c-9aef-a06537692f56" name="Changes"
comment="" />
    <created>1651483182749</created>
    <option name="number" value="Default" />
    <option name="presentableId" value="Default" />
    <updated>1651483182749</updated>
  </task>
  <servers />
</component>
<component name="WindowStateProjectService">

```

```

    <state x="368" y="32" key="FileChooserDialogImpl" timestamp="1651748465092">
      <screen x="0" y="0" width="1280" height="672" />
    </state>
    <state x="368" y="32" key="FileChooserDialogImpl/0.0.1280.672@0.0.1280.672"
timestamp="1651748465092" />
  </component>
</project>
<?xml version="1.0" encoding="UTF-8"?>
<module type="PYTHON_MODULE" version="4">
  <component name="NewModuleRootManager">
    <content url="file://$MODULE_DIR$" />
    <orderEntry type="inheritedJdk" />
    <orderEntry type="sourceFolder" forTests="false" />
  </component>
  <component name="PyDocumentationSettings">
    <option name="format" value="PLAIN" />
    <option name="myDocStringFormat" value="Plain" />
  </component>
</module>
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'OnlineFire_ReportingSystem.settings')
    try:
        from django.core.management import execute_from_command_line

```

```

except ImportError as exc:
    raise ImportError(
        "Couldn't import Django. Are you sure it's installed and "
        "available on your PYTHONPATH environment variable? Did you "
        "forget to activate a virtual environment?"
    ) from exc
execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

from django.contrib import admin
from .models import *
# Register your models here.

admin.site.register(Firereport)
admin.site.register(Firetequesthistory)
admin.site.register(Teams)
from django.apps import AppConfig
class FirereportConfig(AppConfig):
    name = 'firereport'
from django.db import models

# Create your models here.
class Teams(models.Model):
    teamName = models.CharField(max_length=200, null=True)
    teamLeaderName = models.CharField(max_length=250, null=True)
    teamLeadMobno = models.CharField(max_length=15, null=True)
    teamMembers = models.CharField(max_length=300, null=True)
    postingDate = models.DateTimeField(auto_now_add=True)
    #amount = models.DecimalField(max_digits=10, decimal_places=2, null=True,
    blank=True)

```

```
def __str__(self):  
    return self.teamName
```

```
class Firereport(models.Model):
```

```
    FullName = models.CharField(max_length=250, null=True)  
    MobileNumber = models.CharField(max_length=12, null=True)  
    Location = models.CharField(max_length=200, null=True)  
    Message = models.CharField(max_length=200, null=True)  
    AssignTo = models.ForeignKey(Teams, on_delete=models.CASCADE, null=True)  
    Status = models.CharField(max_length=150, null=True)  
    Postingdate = models.DateTimeField(auto_now_add=True)  
    AssignedTime = models.CharField(max_length=150, null=True)  
    UpdationDate = models.DateTimeField(null=True)
```

```
def __str__(self):  
    return self.FullName
```

```
class Firetequesthistory(models.Model):
```

```
    firereport = models.ForeignKey(Firereport, on_delete=models.CASCADE,  
null=True)  
    status = models.CharField(max_length=200, null=True)  
    remark = models.CharField(max_length=250, null=True)  
    postingDate = models.DateTimeField(auto_now_add=True)
```

```
def __str__(self):  
    return self.status
```

```
import datetime
```

```
from django.db.models import Q
```

```
from django.shortcuts import render,redirect
```

```

from .models import *
from datetime import date
from datetime import datetime
from django.contrib.auth.models import User
from django.contrib.auth import login,logout,authenticate
# Create your views here.

def index(request):
    return render(request, 'index.html')

def reporting(request):
    error = ""
    if request.method == "POST":
        FullName = request.POST['FullName']
        MobileNumber = request.POST['MobileNumber']
        Location = request.POST['Location']
        Message = request.POST['Message']
        try:
            Firereport.objects.create(FullName=FullName, MobileNumber=MobileNumber,
Location=Location, Message=Message)
            error = "no"
        except:
            error = "yes"
    return render(request, 'reporting.html', locals())

#Admin New Add
def adminaddnewtender(request):
    error = ""
    if request.method == "POST":
        FullName = request.POST['FullName']
        MobileNumber = request.POST['MobileNumber']

```

```

    Location = request.POST['Location']
    Message = request.POST['Message']
    try:
        Firereport.objects.create(FullName=FullName, MobileNumber=MobileNumber,
Location=Location, Message=Message)
        error = "no"
    except:
        error = "yes"
    return render(request, 'reporting.html', locals())

```

```

def viewStatus(request):
    sd = None
    if request.method == 'POST':
        sd = request.POST['searchdata']
        try:
            firereport = Firereport.objects.filter(Q(FullName__icontains=sd) |
Q(MobileNumber=sd) | Q(Location__icontains=sd))
        except:
            firereport = ""
    else:
        try:
            firereport = Firereport.objects.filter()
        except:
            firereport = ""
    return render(request, 'viewStatus.html', locals())

```

```

def viewStatusDetails(request,pid):
    firereport = Firereport.objects.get(id=pid)
    report1 = Firetequesthistory.objects.filter(firereport=firereport)
    reportcount = Firetequesthistory.objects.filter(firereport=firereport).count()
    return render(request, 'viewStatusDetails.html', locals())

```

```

def admin_login(request):
    error = ""
    if request.method == 'POST':
        u = request.POST['uname']
        p = request.POST['password']
        user = authenticate(username=u, password=p)
        try:
            if user.is_staff:
                login(request, user)
                error = "no"
            else:
                error = "yes"
        except:
            error = "yes"
    return render(request, 'admin_login.html', locals())

def dashboard(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    totalnewrequest = Firereport.objects.filter(Status__isnull=True).count()
    totalAssign = Firereport.objects.filter(Status='Assigned').count()
    totalontheway = Firereport.objects.filter(Status='Team On the Way').count()
    totalworkprocess = Firereport.objects.filter(Status='Fire Relief Work in Progress').count()
    totalreqcomplete = Firereport.objects.filter(Status='Request Completed').count()
    totalfire = Firereport.objects.all().count()
    return render(request, 'admin/dashboard.html', locals())

def addTeam(request):
    if not request.user.is_authenticated:

```

```

        return redirect('admin_login')
error = ""
if request.method == "POST":
    teamName = request.POST['teamName']
    teamLeaderName = request.POST['teamLeaderName']
    teamLeadMobno = request.POST['teamLeadMobno']
    teamMembers = request.POST['teamMembers']
    try:
        Teams.objects.create(teamName=teamName,
teamLeaderName=teamLeaderName,          teamLeadMobno=teamLeadMobno,
teamMembers=teamMembers)
        error = "no"
    except:
        error = "yes"
    return render(request, 'admin/addTeam.html', locals())
#Add Tender by Admin
# def adminaddnewtender(request):
#     if not request.user.is_authenticated:
#         return redirect('admin_login')
#     error = ""
#     if request.method == "POST":
#         FullName = request.POST['FullName']
#         MobileNumber = request.POST['MobileNumber']
#         Location = request.POST['Location']
#         Message = request.POST['Message']

#     try:
#         Teams.objects.create(FullName=FullName, MobileNumber=MobileNumber,
Location=Location, Message=Message)
#         error = "no"
#     except:

```

```

#         error = "yes"
#     return render(request, 'reporting.html', locals())

def manageTeam(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    teams = Teams.objects.all()
    return render(request, 'admin/manageTeam.html', locals())

def editTeam(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    teams = Teams.objects.get(id=pid)
    error = ""
    if request.method == "POST":
        teamName = request.POST['teamName']
        teamLeaderName = request.POST['teamLeaderName']
        teamLeadMobno = request.POST['teamLeadMobno']
        teamMembers = request.POST['teamMembers']

        teams.teamName = teamName
        teams.teamLeaderName = teamLeaderName
        teams.teamLeadMobno = teamLeadMobno
        teams.teamMembers = teamMembers

    try:
        teams.save()
        error = "no"
    except:
        error = "yes"

```

```

return render(request, 'admin/editTeam.html', locals())

def deleteTeam(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    teams = Teams.objects.get(id=pid)
    teams.delete()
    return redirect('manageTeam')

def newRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status__isnull=True)
    return render(request, 'admin/newRequest.html', locals())

def assignRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='Assigned')
    return render(request, 'admin/assignRequest.html', locals())

def teamontheway(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='Tender Assigned')
    return render(request, 'admin/teamontheway.html', locals())

def workinprogress(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='In Progress')

```

```

return render(request, 'admin/workinprogress.html', locals())

def completeRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='Approved')
    return render(request, 'admin/completeRequest.html', locals())

def allRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.all()
    return render(request, 'admin/allRequest.html', locals())

def new_Request(request):
    firereports = Firereport.objects.all()
    return render(request, 'newRequest.html', {'firereports': firereports})

def deleteRequest(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.get(id=pid)
    firereport.delete()
    return redirect('allRequest')

def viewRequestDetails(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.get(id=pid)
    report1 = Firetequesthistory.objects.filter(firereport=firereport)

```

```

firereportid = firereport.id
team = Teams.objects.all()
reportcount = Firetequesthistory.objects.filter(firereport=firereport).count()
try:
    if request.method == "POST":
        teamid = request.POST['AssignTo']
        Status="Assigned"
        team1 = Teams.objects.get(id=teamid)
        try:
            firereport.AssignTo = team1
            firereport.Status = Status
            now = datetime.now()
            firereport.AssignedTime = now.strftime("%m/%d/%Y %H:%M:%S")
            firereport.save()
            error = "no"
        except:
            error = "yes"
    except:
        if request.method == "POST":
            status = request.POST['status']
            remark = request.POST['remark']

            try:
                requesttracking =
                Firetequesthistory.objects.create(firereport=firereport,status=status, remark=remark)
                firereport.Status = status
                firereport.save()
                firereport.UpdationDate = date.today()
                error1 = "no"
            except:
                error1 = "yes"

```

```

return render(request, 'admin/viewRequestDetails.html', locals())

def dateReport(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    error = ""
    if request.method == 'POST':
        fd = request.POST['fromDate']
        td = request.POST['toDate']
        firereport = Firereport.objects.filter(Q(Postingdate_gte=fd) &
Q(Postingdate_lte=td))
        return render(request, 'admin/betweendateReportDtls.html', locals())
    return render(request, 'admin/dateReport.html', locals())

def search(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    sd = None
    if request.method == 'POST':
        sd = request.POST['searchdata']
        try:
            firereport = Firereport.objects.filter(Q(FullName_icontains=sd) |
Q(MobileNumber=sd) | Q(Location_icontains=sd))
        except:
            firereport = ""
    return render(request, 'admin/search.html', locals())

def changePassword(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    error = ""

```

```

user = request.user
if request.method == "POST":
    o = request.POST['oldpassword']
    n = request.POST['newpassword']
    try:
        u = User.objects.get(id=request.user.id)
        if user.check_password(o):
            u.set_password(n)
            u.save()
            error = "no"
        else:
            error = 'not'
    except:
        error = "yes"
    return render(request, 'admin/changePassword.html', locals())

def Logout(request):
    logout(request)
    return redirect('index')

```

B. SCREEN SHOTS



Fig: B.1 Home Page

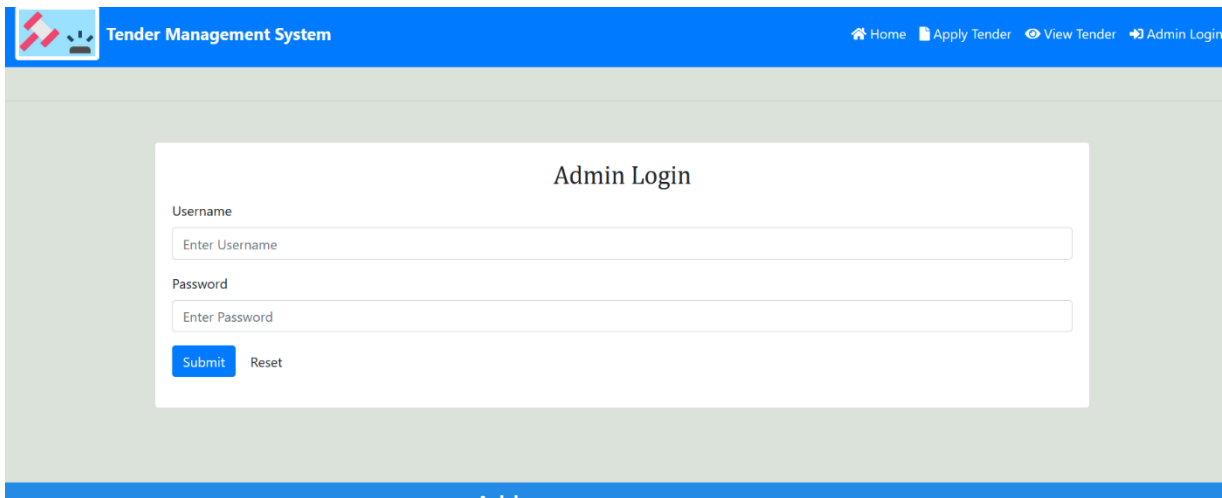


Fig: B.2 Admin Login

**Tender Management System**

[Home](#) [Apply Tender](#) [View Tender](#) [Admin Login](#)

APPLY TENDOR ONLINE

BID FOR THE TENDER

Buyer Name

Enter Name

Mobile Number

Enter Mobile No.

Location

Enter Your Location

Company Name, Bid-Amount & Details

Use Comma to seprate Details....

Reset

Bid Now

Fig: B.3 Apply for Tender

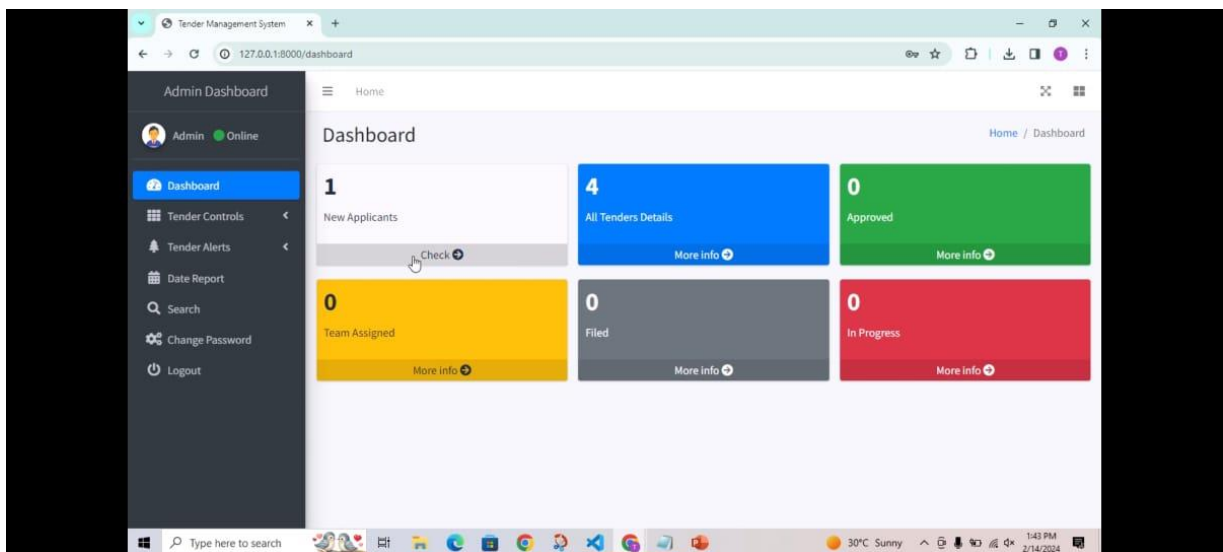


Fig: B.4 Admin Dashboard

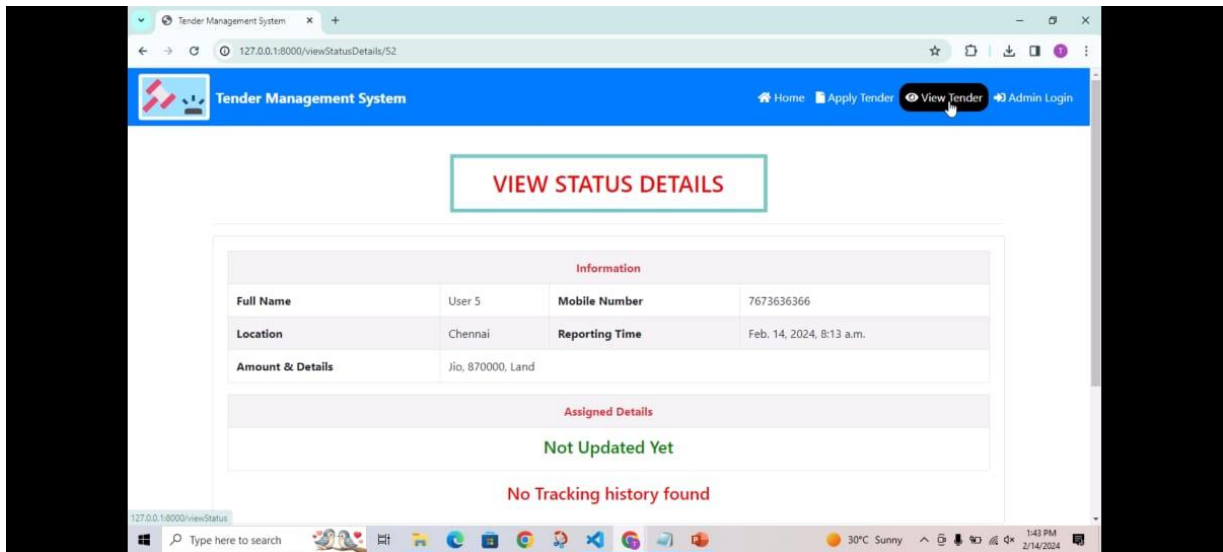


Fig: B.5 Tender Status Details

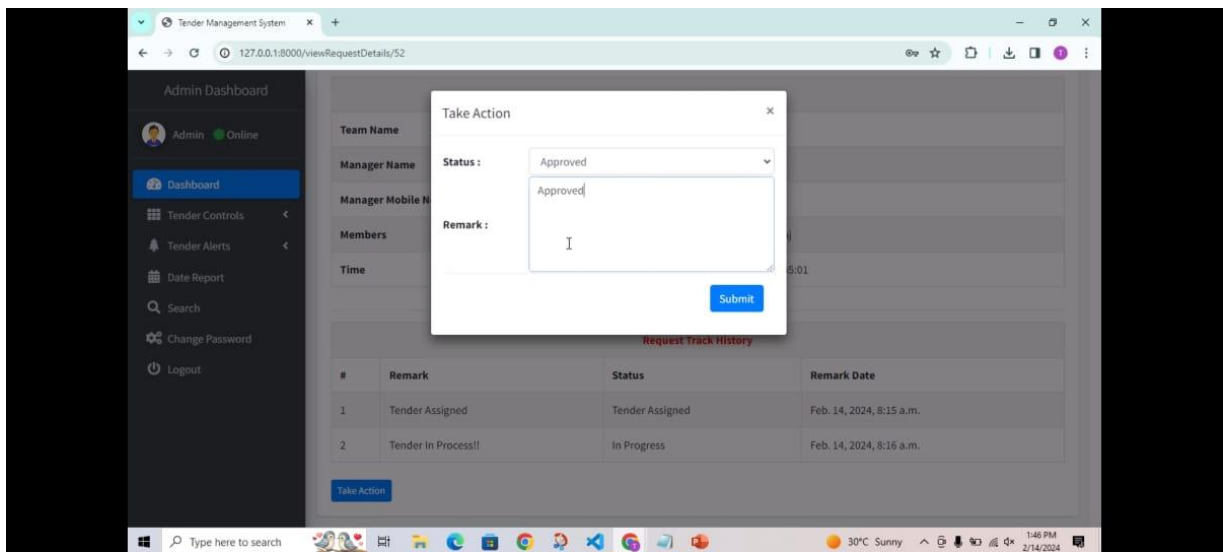


Fig: B.6 Tender Action Page

C.RESEARCH PAPER

SMART TENDER CONTRACT MANAGEMENT SYSTEM IN PYTHON USING BLOCKCHAIN

Srinivasa Rao, Venkata Bhaskar and Mrs.Sageengrana

Sathyabama Institute of Science and Technology, Chennai, India

a)narisetty.srinivasarao77@gmail.com

b)lovelybhaskar.10.com@gmail.com

c)sageengranadhas@gmail.com

ABSTRACT- Supply chain communication is a key hobby that facilitates corporations find the right human beings to hire. In structure and development corporations, customers lease individuals who calculate and provide estimates of the expenses related to constructing of entirety. Thus, the purchaser could make his choice primarily based on the end result of tenders to pick out the most suitable challenge contractor. Donors need to acquire the e- book manager, compare it and produce it to this office before being notified. In this paper, we gift the layout and implementation of an electronic procurement system the use of internet-primarily based control as a mechanism for such furnish activities. In this paper, we've got used the innovation of a complicated block chain structure to overcome security demanding situations and meet infrastructure necessities. A comfy structure of privilege in any union or U.S.A. Permits new freedoms to be expressed. This notion is higher than other schemes, due to the fact it is a decentralized structure that operates a spot with exclusive currencies at stake. We additionally need to interrupt down the security of our proposed grant shape, which indicates that our meeting is secure inside the event of conflicts with numerous agreements

Smart Tender Contract Management System (STCMS) is a novel application developed in Python leveraging blockchain technology for enhanced transparency, security, and efficiency in tender contract management processes. Traditional tender contract management systems often face challenges such as lack of transparency, susceptibility to fraud, and manual errors. STCMS addresses these issues by integrating smart contracts on a blockchain platform.

This paper presents the design and implementation of STCMS, focusing on its key features and functionalities. The system allows stakeholders, including buyers, suppliers, and evaluators, to interact securely and transparently throughout the tendering process. Smart contracts are used to automate various aspects of tender contract management, including bid submission, evaluation, and contract execution.

The use of blockchain technology ensures that all transactions and interactions within the system are recorded immutably on the distributed ledger, providing a tamper-proof audit trail. Additionally, cryptographic techniques are employed to ensure data privacy and integrity.

INTRODUCTION

Existing e-trade structures are not "truthful and open", that means that statistics isn't shared with all people. Stakeholders (right to Information). Information

For example, it's far published "at its discretion" - whilst a firm is selected because the winner of the agreement, other companies participating within the gentle are not informed why their bids have been rejected. ? , a success corporation can request this records, but obtaining this information is a complex technique. Although it's far viable to hear those files, it is still time to assess them. Without simplicity, safety is a enormous difficulty of these portals, which results in the distortion and control of the records accrued within the information. If a hacker gains manipulate of this centralized database, it can be leaked to competitors, ensuing in huge financial and strategic commercial enterprise losses. The innovation barrier to those security features may be met because its miles based on decentralization and statistics encryption with an impregnable quadratic boom of alternate directors. Thus, the block chain can be used as a transparent, decentralized and comfortable gadget of tenders, which allows to control the

charges of its gates and display the operations of all of the sports finished via the gentle gates.

Current e-commerce systems often lack transparency and openness, leading to limited access to information for stakeholders. This lack of transparency can be particularly problematic in tendering processes, where decisions are made without clear justification and unsuccessful bidders are left in the dark about the reasons for their rejection.

For instance, when a firm is awarded a contract, other participating companies may not receive detailed feedback on why their bids were unsuccessful. While successful firms have the right to request this information, the process of obtaining it can be complex and time-consuming. This lack of simplicity not only hampers transparency but also raises concerns about data security within these systems.

The centralized nature of existing e-commerce platforms poses significant security risks, as a breach in the system could lead to the leakage of sensitive information to competitors, resulting in substantial financial and strategic losses for businesses. Overcoming these security challenges requires innovative solutions that prioritize decentralization and data encryption to ensure the integrity and confidentiality of information.

OBJECTIVES

We use modern-day generation to resolve soft strength troubles and make sure safety and stability.

LITERATURE REVIEW

Literature review is an important step in the software program improvement method. Before the tool is evolved, it's miles important to determine the time factors, budget and strengths of the agency. With all these conditions met, the following step is to determine which running system and language can be used to increase the tool. When programmers begin developing a tool, they may want extra help. This assist can be obtained from senior packages, books or websites. Before designing the machine, the above factors are saved in mind while designing the machine. The maximum important part of the paintings of the venture improvement department is the specified evaluation of all the necessities which can be important to broaden the mission. Literature assessment is the maximum essential area of application improvement for any venture. Before developing the equipment and the related plan, it's far important to discover and analyze the factors of time, aid requirements, manpower, monetary and organizational strengths. After a lot of these questions were replied and the studies carried out, the next step is to determine the specification of the software in the pc mission, what form of operating system will be required for the reason, and what software might be required for the challenge. Work Making tracks like equipment and associated sports.

1. Intelligent software tools the use of Ethereum.

By growing a software for the Ethereum block chain that can be used to control software program. Smart agreement facilitates in constructing the software as per the guidelines and conditions of the corporation. In addition, this transaction, applied at the Ethereum block chain, makes the utility reliable, getting rid of the involvement of 0.33 parties. The application can be accessed and downloaded by using exclusive users, and the data isn't always saved in one vicinity. This decentralization is finished via disbursed belongings at the Ethereum block chain. Casper on Ethereum has verified powerful for verifying all varieties of transactions inside the software. In Ethereum, the various types of orders created for a particular reason of the software are effectively represented and managed. When comparing Hyper ledger and Ethereum, we used Ethereum as it is basically a consumer-facet application. Different development groups can use it to create programs and share rewards through collaboration.

2. Decentralized Market Application on Block chain

Modern, centralized on-line markets like eBay provide consumers with an opportunity choice to without problems sell and buy merchandise. However, the risks of these markets consist of the capacity of the platform to block dealers at will, the charges that must be paid to the platform when listing and promoting the product and the lack of person privacy statistics. In this article, we advocate an software that addresses all3 of those shortcomings the use of the Ethereum block chain platform. The application become advanced the use of a TUBE development surroundings. The feature of the application is

contained in an ethereal smart settlement, that's then broadcast on the ethereal network. User enter is read via the net interface and dispatched to the ethereal community via the Web3.js API. Statistics had been accumulated using the Rinke by take a look at network. The common application transaction time changed into proven to be 3.Eight seconds, and the common fuel intake changed into 4.6 VE. The contract creation time for the application is proven in less than a 2nd. After this the cost evaluation turned into carried out. Gas intake for corporations to shop for and promote gadgets is transformed to US greenbacks, and the gas price for the app is compared to the cost of using an public sale website online along with eBay and in-character auctions. Like Sotheby's homes Results show that selling via an app is inexpensive than existing on-line options and present in-individual alternatives. These tests display that our software efficiently addresses the failure of our public sale market.

3. Smart Contracts: Methods for Ecosystem Security and Sustainability.

Smart agreement technology built at the block chain are attracting quite a few attention in new enterprise programs and the medical network, because they allow untrusted events to explicit the terms of the settlement in software program code, eliminating the want for a depended on third party. The method of writing properly-functioning and comfy contracts on ethereal, modern-day most famous smart contract platform, is a daunting mission. Research in this topic has lately started in enterprise and academia. Based on the analysis of the accumulated information the use of idea-based totally methods, we've developed numerous fashionable security standards primarily based on solidity, the principle ethereal programming language. Templates provide answers to explain common safety issues and can be utilized by developers to mitigate not unusual attack eventualities.

4. Design Patterns for Smart Contracts

The concept of digitally simplifying agreement law and enterprise practices the usage of computer software program has given upward push to the idea of clever contracts. Today, the most distinguished patron agreement surroundings is ethereal, a block chain-primarily based, distributed computing platform. Writing legitimate and at ease smart contracts for ethereal is tough, due to the inherent nature of block chain-based contract execution, low programming abstraction, and the regular evolution of systems and security issues. Based on interdisciplinary literature research and facts evaluation accumulated through qualitative research techniques, we recognized numerous layout patterns that provide design recommendations. We describe those patterns in detail and provide pattern code to higher illustrate them. Our studies suggests that the models are extensively used for software necessities and common problems. We wish to generalize some or all the smart settlement models to other ecosystems, however that is outdoor the scope of this look at, which simply examines clever settlement methods in ethereal.

5. Security and trust in blockchain-based system

Security and trust are critical aspects of blockchain-based systems, particularly in the context of tender contract management where sensitive information and financial transactions are involved (Swan, 2015). Studies have explored various security mechanisms such as cryptographic techniques, consensus algorithms, and access control mechanisms to ensure the integrity and confidentiality of data (Yli-Huoma et al., 2016)

6. case studies and practical applications

Several case studies and practical applications have demonstrated the feasibility and benefits of implementing blockchain-based solutions in procurement and tender management (Böhme et al., 2015; Xu et al., 2017). These studies showcase real-world examples of organizations leveraging blockchain to enhance transparency, reduce costs, and improve trust in procurement processes

EXISTING SYSTEM

Lockdown has come to be an inevitable destiny in corporate finance, especially in commercial enterprise operations and logistics. Introducing a similar clever settlement concept, the Trust Sub-Ledger Operation (TSLO), this paper proposes a holistic block chain- primarily based framework to offer traceability and trustworthiness of documents throughout a pool of property. TSLO is a very flexible and adaptable approach for coping with things in accounting and making plans useful resource systems. This method is primarily based at the Decentralized Micro service Tree (DMST) and an extensible shape of digital auction TEA (Triple Accounting). Instead of the usage of a multi-ledger structure, the skeleton of Hyper ledger Fabric, most effective for collaborating channels within a corporation or enterprise, our method makes use of decentralized sub-ledgers with processing bushes

(TMST) for asset-based totally transactions. In addition, the government's method for auditing and taxing financial businesses will offer extra authority and protect more events via a aggregate of proof of authority and proof of possession.

Dis-Advantages

- Smart financial procedures are almost not possible to exchange, and solving each malicious program in the application is time-ingesting and high priced.
- Although there are continually twists and turns in financial affairs, not usually smart creditors can address the twists and turns.

PROPOSED SYSTEM

Currently, corporations consisting of agriculture, healthcare, economic offerings and the general public sector are beginning to make the leap. Although this gentle element remains hindered with the aid of this new era. If soft control is going wrong, abuses will cause massive losses. These include favoritism closer to contractors, terrible report retaining, loss of transparency, hacking, conversion of information and other comparable issues.

We use technology to clear up the problems of gentle control and to ensure that it enables to maintain protection and performance. Thanks to encryption mixed with an impenetrable block architecture, block chain transaction management is absolutely comfy.

Here we use block chain generation to attain project primarily based documents like tender documents, bids, bids, team details, beyond facts, authority approvals info and rejection information.

Enhanced Trust and Integrity: By utilizing blockchain technology for tender contract management, the proposed system enhances trust and integrity within the procurement process. The transparent and immutable nature of the blockchain ledger ensures that all transactions and actions taken during the tendering process are verifiable and tamper-proof, thereby reducing the risk of fraud or corruption.

Improved Compliance and Auditability: The use of blockchain technology enables the system to maintain a comprehensive and auditable record of all tender-related activities. This not only ensures compliance with regulatory requirements but also facilitates easier auditing and accountability for stakeholders involved in the tendering process.

Efficient Dispute Resolution: In the event of disputes or discrepancies, the blockchain-based system provides a reliable and transparent mechanism for resolving conflicts. Since all transactions are recorded on the blockchain in a decentralized manner, disputes can be resolved more efficiently by referring to the immutable audit trail maintained by the system.

Global Accessibility and Inclusivity: The blockchain-based system offers global accessibility and inclusivity, allowing stakeholders from diverse geographical locations to participate in tendering processes seamlessly. This opens up opportunities for small and medium-sized enterprises (SMEs) and vendors from emerging markets to compete on a level playing field with larger corporations.

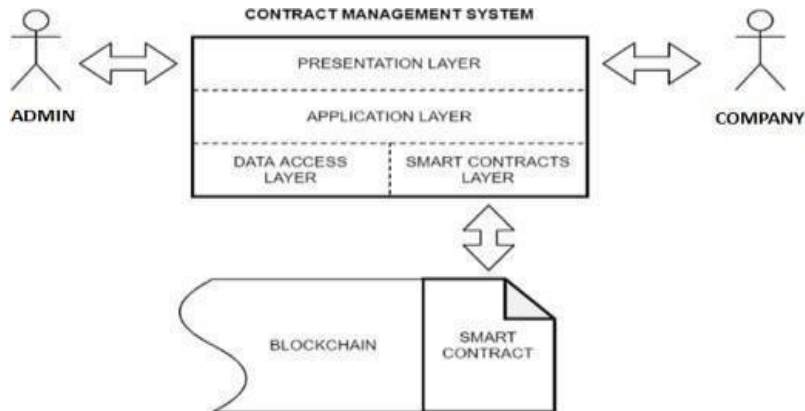
Advantages:

An smart gentle management machine by way of

0.33 parties reduces the possibility of manipulation. Pain tenders although the price is no intermediary. Encryption and cryptography protects all documents from unauthorized access.

Automate laptop systems the usage of protocols, thereby saving time in performing many business strategies.

SYSTEM ARCHITECTURE



SYSTEM REQUIREMENTS

Hardware Requirements System: Pentium i3 processor. Hard Drive: 500 GB.

Monitor: 15" LED

Input tool: keyboard, mouse RAM: 2GB

Software Requirements Operating device: Windows 10 Coding Language: Java

MODULES

- ❖ Admin Module
- ❖ Company Module

Admin:

- log in
- Manage tenders
- List of all tenders
- View / Update / Delete / Add
- add tender
- Update tenders
- Only certain essential and miscellaneous fields can be up to date.
- Tender present day
- Open a list of all tenders for packages/proposals.

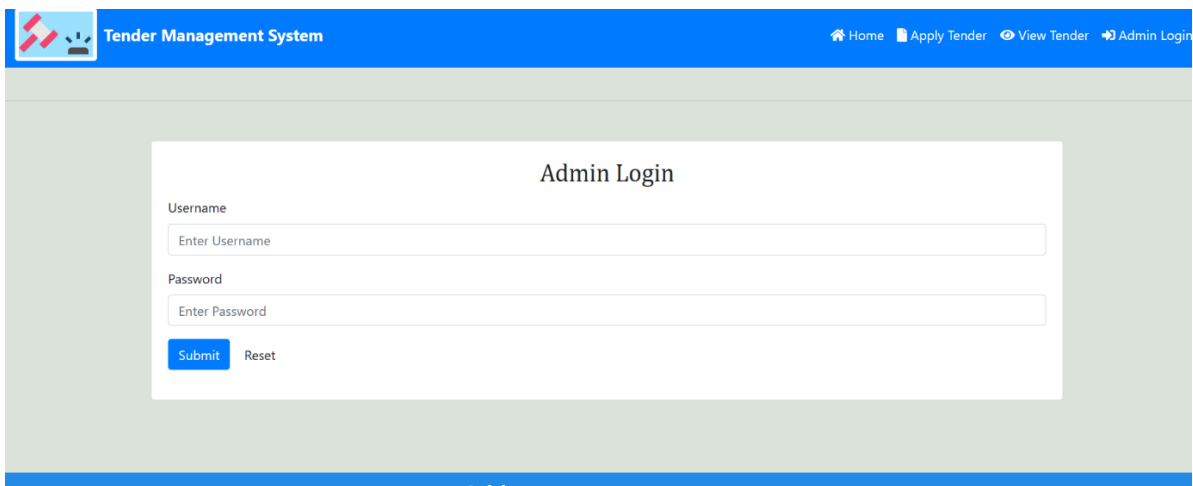
- List of all programs obtained for tenders.
- Administrator replace by way of application fame (Approved / Rejected / Shortlisted)
- Previous tenders
- View packages as a list of all tenders for order status and current and closed renewal.
 - View transactions
- List of tenders and programs for them with processing status
- It will display if a manipulation changed into performed without the consent or knowledge of the administrator.
- At the quit we are able to see how we are able to broaden the treatment.

Company;

- Registration
- log in
- Company is registered
- Tenders
- List of all tenders open to the government
- The enterprise can participate in its proposed undertaking.
- alternate the password
- Go to the admin panel, view the application and alternate the reput.
- Let's visit the "Companies" group.
- My utility
- List of packages and standing

RESULTS

We use technology based totally transaction documents to secure smooth files, bids, bids, team info, past records, approval information and rejection details.



Tender Management System

Home Apply Tender View Tender Admin Login

Admin Login

Username

Password

[Submit](#) [Reset](#)

- If the user already has account, he directly login

The screenshot shows the 'BID FOR THE TENDER' form within the Tender Management System. The form includes input fields for Buyer Name, Mobile Number, Location, and Company Name/Bid Amount & Details. A 'Reset' button is on the bottom left, and a 'Bid Now' button is on the bottom right. The header bar contains links for Home, Apply Tender, View Tender, and Admin Login.

- This image shows how to apply for the tender

The screenshot shows the search and tenders list section of the Tender Management System. It features a search bar with the placeholder 'Enter Name/Mobile No/Location' and a search icon. Below the search bar is a table titled 'Tenders to Apply'.

#	Tender Name	Mobile Number	Location	Uploaded Time	Status
1	Sample	1234567890	Chennai	Feb. 13, 2024, 7:49 a.m.	
2	User1	8736366363	Ramnad	Feb. 13, 2024, 8:45 a.m.	
3	User3	6363563536	Chennai	Feb. 13, 2024, 9:19 a.m.	
4	User 5	7673636366	Chennai	Feb. 14, 2024, 8:13 a.m.	
5	syed	2142345	chennai	March 22, 2024, 12:02 p.m.	

- This image shows the tender to apply

Future Scope

Looking ahead, the smart tender contract management system developed in Python using blockchain technology holds immense potential for further advancements and future scope. One promising area for future exploration involves expanding the system's capabilities to support a broader range of procurement processes beyond tender contract management. This could include functionalities for supplier management, contract lifecycle management, and procurement analytics, allowing organizations to streamline their entire procurement workflow on a single platform. Additionally, there is an opportunity to enhance the system's scalability and interoperability to accommodate larger transaction volumes and integrate seamlessly with other blockchain networks and external systems. Furthermore, future iterations of the system could incorporate advanced features such as AI-driven

decision support, predictive analytics, and autonomous contract execution, enabling organizations to make data-driven decisions and automate routine procurement tasks. Moreover, there is a growing need for blockchain-based solutions that address sustainability and ethical sourcing concerns in procurement, presenting an exciting avenue for future development. By integrating sustainability metrics, supply chain transparency, and ethical sourcing criteria into the system, organizations can ensure responsible and socially conscious procurement practices. Overall, the future scope for the smart tender contract management system in Python using blockchain technology is vast, encompassing advancements in functionality, scalability, interoperability, sustainability, and automation to meet the evolving needs of modern procurement practices

CONCLUSION

When it involves packages like tender control, security and transparency are vital, traditional technology and information planning strategies can no longer be used as they pose the risk of records corruption. As we realize, delicate systems using centralized databases to make contracts and make orders have many security requirements. SHA and AES algorithms for encryption and the usage of NLP principles to assess patron remarks enables the smooth enterprise to choose the great proposition within the listing. The use of generation including block chain, decentralized databases, can provide the safety and openness vital for these sorts of programs. This article discusses how any such machine can be designed, deliberating the diverse approaches involved, and their underlying implementation.

REFERENCES:

- "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967. K. C. Davis, "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967.
 - The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967, "The Information Act: A Preliminary Analysis." "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967. K. C. Davis, "The Information Act: A Preliminary Analysis," The University of Chicago Law Review, vol. 34, no. 4, pp. 761–816, 1967.3. Pal, Om, and Surendra Singh.
 - Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin Betts, Martin 89-102 in Journal of Information Technology in Construction, vol. 11 (2006).
 - Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE international congress on big data (BigData congress). IEEE, 2017.
- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.
 - Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media.
 - Tapscott, D., & Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World. Penguin.

- Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4, 2292-2303.
- Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *IEEE International Congress on Big Data* (pp. 557-564). IEEE.
- Swan, M. (2015). *Blockchain: Blueprint for a New Economy*. O'Reilly Media.
- Lewis, G. A., Mavridou, T., & Gorbach, A. (2018). A Blockchain-Based Smart Contract System for Healthcare Management. In *IEEE International Conference on Software Architecture (ICSA) Workshops* (pp. 230-233). IEEE.
- Zheng, Z., Xie, S., Dai, H. N., & Wang, H. (2017). Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, 14(4), 352-375.
- Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, 14(4), 352-375.
- Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaria, V. (2018). Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?. *Future Internet*, 10(2), 20.

RE-2022-230327-plag-report

ORIGINALITY REPORT

6%

SIMILARITY INDEX

5%

INTERNET SOURCES

3%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

www.researchgate.net

Internet Source

1%

2

jpinfotech.org

Internet Source

1%

3

ijact.org

Internet Source

1%

4

[Submitted to University of West London](#)

Student Paper

1%

5

[Submitted to Liverpool John Moores University](#)

Student Paper

<1%