A

Major Project

On

BLOCKCHAIN FOR THE MANAGEMENT OF INTERNET OF THINGS DEVICES IN THE MEDICAL INDUSTRY

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

P Indhu	(217R1A0544)
K Mahipal	(217R1A0528)
E Mahesh Kumar	(217R1A0520)

Under the Guidance of

Ms. TABEEN FATIMA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

April, 2025.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
TECHNICAL CAMPUS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled “**BLOCKCHAIN FOR THE MANAGEMENT OF INTERNET OF THINGS DEVICES IN THE MEDICAL INDUSTRY**” being submitted by **P. Indhu (217R1A0544), K. Mahipal (217R1A0528) & E. Mahesh Kumar (217R1A0520)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Ms. Tabeen Fatima
Assistant Professor
INTERNAL GUIDE

Dr. Nuthanakanti Bhaskar
HoD

Dr. A. Raji Reddy
DIRECTOR

Signature of External Examiner

Submitted for viva voice Examination held on ______________________

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Ms. Tabeen Fatima,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We take this opportunity to extend our heartfelt appreciation to the Project Review Committee (PRC) Coordinators—**Dr. K. Maheswari, Dr. J. Narasimharao, Ms. K. Shilpa, and Mr. K. Ranjith Reddy**—for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

Our sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of our project.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

We also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, we sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

P Indhu (217R1A0544)
K Mahipal (217R1A0528)
E Mahesh Kumar (217R1A0520)

VISION AND MISSION

INSTITUTE VISION:

To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

INSTITUTE MISSION:

1. To create state of art facilities for effective Teaching- Learning Process.
2. Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.
3. Infuse Professional, Ethical and Societal values among Learning Community.

DEPARTMENT VISION:

To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

DEPARTMENT MISSION:

1. To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.
2. To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student’s practical knowledge.
3. To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

ABSTRACT

This project is titled as “Blockchain For The Management Of Internet Of Things Devices In The Medical Industry”. With the recent emergence of the smart healthcare era, and patients relying more on personalized health monitoring based on Internet of Medical Things (IoMT) devices; patients’ lives are becoming highly threatened in case they fall victim to counterfeit devices. Thus, verifying whether these body sensors utilized are authentic and reliable in an unimpeachable, credible, and auditable manner without any centralized management is of crucial importance. Furthermore, manipulating data and hijacking in an IoMT context are also of tremendous criticality. Motivated by the aforementioned challenges, in this article, a smart contract based scalable authentication scheme dedicated for IoMT devices is proposed. The scheme mitigates the deficiencies of the traditional established systems that are extensively built on centralized approaches, vulnerable to distributed denial of service attacks, by leveraging blockchain’s decentralization and security properties. The scheme ensures confidentiality, anonymity, and privacy as it is built on a consortium blockchain and integrity by offering secure firmware updates and protects patients from counterfeit devices by leveraging the physical unclonable function. The rapid advancement of Internet of Things (IoT) technology in healthcare has resulted in the widespread deployment of connected medical devices, which enhance patient monitoring and treatment. However, these advancements raise significant challenges in data security, privacy, and interoperability. This study presents a blockchain based framework designed to manage IoT devices in the medical industry. By utilizing blockchain’s decentralized architecture, the proposed system ensures secure data transmission, enhances patient privacy, and provides a transparent audit trail for all interactions involving medical devices. Additionally, the framework addresses interoperability issues by standardizing data formats and communication protocols. The results demonstrate that blockchain technology not only enhances the security of IoT devices but also fosters collaboration among healthcare providers, ultimately leading to improved patient outcomes.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

The project, titled " Blockchain For The Management Of Internet Of Things Devices In The Medical Industry" is designed to develop an integration of blockchain technology within the management of Internet of Things (IoT) devices in the medical industry represents a groundbreaking advancement in healthcare. As IoT devices proliferate in medical settings ranging from wearable health monitors to connected surgical instruments there arises an urgent need for secure, efficient, and transparent data management. Blockchain, with its decentralized and immutable ledger capabilities, offers a robust solution to the challenges of data integrity, interoperability, and patient privacy. By facilitating secure data sharing among various stakeholders, including healthcare providers, patients, and device manufacturers, blockchain technology can enhance the accuracy of medical records, streamline operational processes, and foster trust in automated systems. This convergence of IoT and blockchain not only promises improved patient outcomes through real time monitoring and analytics but also sets the stage for innovative healthcare solutions that prioritize data security and patient empowerment in an increasingly interconnected world.

1.1 PROJECT PURPOSE

The project "Blockchain for the Management of Internet of Things (IoT) Devices in the Medical Industry" aims to address the growing challenges related to security, privacy, data integrity, and device management in healthcare systems. As IoT devices play an increasingly vital role in patient monitoring and medical data collection, the need for secure, transparent, and efficient management becomes critical. Blockchain technology offers a solution by providing a decentralized, immutable ledger for storing and securing the data generated by IoT devices, ensuring data integrity and preventing tampering.

Blockchain enables the seamless interoperability of different IoT devices and systems, allowing them to share data effectively and in real time. With the integration of smart contracts, the management of IoT devices can be automated, reducing the risk of human error and system failures while improving efficiency.

The project ensures transparency and auditability, allowing healthcare providers and regulators to easily track and verify data usage. Ultimately, by leveraging blockchain for IoT device management, the project aims to improve patient outcomes, streamline healthcare processes, reduce costs, and ensure compliance with regulatory standards.

1.2 PROJECT FEATURES

This project incorporates several key features to improve the security, efficiency, scalability, and overall management of IoT devices in the medical industry, improving both the quality of care and operational effectiveness within healthcare systems.

Decentralized Data Storage and Management: The project utilizes blockchain technology to provide decentralized and immutable data storage for IoT devices in the medical industry. Data generated by IoT devices, such as patient monitoring information or sensor readings, is securely stored on a blockchain ledger. This decentralized approach reduces reliance on centralized systems, which can be vulnerable to attacks or failures. By using blockchain, the medical industry can ensure that critical health data is safely distributed across multiple nodes, ensuring redundancy and improving overall data security.

Enhanced Security and Data Integrity: Blockchain’s cryptographic techniques ensure that the data generated by IoT devices remains secure and tamper-proof. The data stored on the blockchain is encrypted, and its integrity is maintained through cryptographic hashes, making it nearly impossible to alter or manipulate once recorded.

Smart Contract Automation: Smart contracts are a key feature of blockchain integration in this project. These self-executing contracts automatically enforce the terms and conditions agreed upon by different parties in a healthcare system. Smart contracts automate various processes, including device authentication, data transmission, and the execution of predefined actions based on data from IoT devices.

By integrating these features, this blockchain provides a security, efficiency, scalability, and overall management of IoT devices in the medical industry

2.LITERATURE SURVEY

The integration of blockchain technology with Internet of Things (IoT) devices in healthcare has garnered significant attention as a solution to some of the most pressing challenges faced by the medical industry, such as data security, privacy, and device management. As IoT devices play an increasingly pivotal role in monitoring patient health and gathering critical data, ensuring that this data is securely managed and communicated is paramount. Several studies have explored the potential of blockchain to enhance the security, integrity, and interoperability of healthcare IoT devices, providing a comprehensive framework for addressing these challenges.

Blockchain's ability to provide a decentralized, tamper-proof, and transparent ledger system has positioned it as a promising technology for addressing the security and privacy concerns inherent in managing medical data. In *MedRec: Using Blockchain for Medical Data Access and Permission Management*, Azaria et al. (2016) proposed the application of blockchain to secure medical records, focusing on ensuring data integrity and patient privacy. Their work highlights blockchain's role in overcoming the shortcomings of traditional centralized systems, such as vulnerability to data breaches and unauthorized access. Similarly, Mettler (2016), in his paper *Blockchain Technology in Healthcare: The Revolution Starts Here*, explored the potential of blockchain for managing electronic health records (EHRs), emphasizing how blockchain technology can facilitate secure patient data exchanges while maintaining privacy. The consensus in these studies is that blockchain offers a reliable solution for securing sensitive patient information, making it an ideal tool for the healthcare sector.

In addition to securing data, blockchain also plays a crucial role in the management of IoT devices within healthcare systems. Xu et al. (2018), in their work titled *Blending Blockchain and Internet of Things for Secure and Scalable E-Health Services*, explored how blockchain could enhance the management of IoT devices by ensuring secure authentication, device status tracking, and data transmission. With IoT devices increasingly being used to monitor patient health, the ability to securely manage and authenticate these devices is essential to preventing unauthorized access or manipulation of device data. Blockchain's decentralized nature eliminates the need for a central authority, reducing the risk of single points of failure. The research conducted by Swan (2015), in her influential book *Blockchain: Blueprint for a New Economy*, also supports the notion that blockchain can create a trusted and secure network for IoT devices, allowing them to interact without compromising the integrity of patient data.

One of the significant challenges in healthcare IoT systems is ensuring interoperability between devices from different manufacturers. Tremblay et al. (2017), in *Improving IoT Interoperability with Blockchain-based Middleware Solutions*, proposed that blockchain could facilitate interoperability by acting as a universal platform for IoT devices to securely communicate with each other and share data. This is crucial in healthcare settings, where devices from various manufacturers need to seamlessly interact in real-time. Blockchain's decentralized structure ensures that data can be securely exchanged without the need for centralized intermediaries, promoting collaboration between different medical systems and devices. This ability to ensure smooth interoperability between diverse IoT devices could lead to improved healthcare delivery and better patient outcomes by allowing for more efficient data sharing.

The issue of security is also paramount when it comes to real-time data collection and monitoring in healthcare. Khan et al. (2018), in their paper *Secure and Trustworthy Smart Healthcare Using Blockchain Technology*, identified the vulnerabilities of IoT devices, especially in the healthcare sector, where data breaches and unauthorized access can lead to severe consequences. Blockchain's encryption and cryptographic techniques provide a strong safeguard against these risks. By ensuring the integrity of data and preventing unauthorized alterations, blockchain can secure the real-time data collected by medical devices. This data can then be used to support timely decision-making by healthcare professionals, ultimately improving patient care and safety. Zhang et al. (2019), in *Towards Secure and Privacy-Preserving Data Sharing in eHealth Systems via Blockchain*, further explored how blockchain can store and manage real-time health data from wearable IoT devices, ensuring that the data is validated and accurate before being used for medical decisions.

Blockchain also provides significant benefits when it comes to regulatory compliance and patient data privacy. In healthcare, strict regulations such as HIPAA and GDPR govern how medical data should be handled and protected. Blockchain's transparency and immutability make it an ideal tool for ensuring compliance with these regulations. Petersen et al. (2019), in *Blockchain-Based Secure Data Sharing for EHR Systems*, suggested that blockchain can provide an auditable and transparent record of who accessed a patient's data and when, allowing healthcare providers to demonstrate their adherence to privacy regulations. Blockchain can also empower patients to have greater control over their data. By providing a decentralized system, blockchain enables patients to selectively share their medical information, maintaining their privacy while allowing access to authorized individuals. This aligns with the growing demand for patient-centered care and enhanced data privacy in healthcare.

The use of smart contracts in blockchain-based IoT systems offers the potential to automate processes and workflows within healthcare settings. Smart contracts can trigger predefined actions when certain conditions are met, such as alerting healthcare providers when a device malfunctions or when abnormal data is recorded. Patel et al. (2018), in *A Review of Smart Contract Applications in Healthcare*, highlighted the potential of smart contracts to streamline healthcare workflows, reduce human error, and improve the efficiency of device management. By automating routine tasks, smart contracts can help ensure that critical actions are taken promptly, reducing delays and improving the overall quality of care.

The scalability of blockchain technology is another significant advantage in managing healthcare IoT devices. As the number of IoT devices in healthcare settings continues to grow, it is essential for the system to scale efficiently without compromising performance. Zheng et al. (2018), in their comprehensive survey *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*, noted that blockchain's ability to decentralize the management of IoT devices could enable healthcare systems to scale seamlessly as the number of devices increases, ensuring continued efficiency and security as more devices are integrated into the system.

2.1 REVIEW OF RELATED WORK

The integration of blockchain with Internet of Things (IoT) devices in healthcare has been widely explored to address issues related to data security, device management, and interoperability. **Azaria et al. (2016)** and **Mettler (2016)** explored blockchain's potential for secure medical data management, highlighting its ability to ensure data integrity and patient privacy by providing a decentralized, tamper-proof ledger. This is critical for managing Electronic Health Records (EHRs) and preventing unauthorized access.

Xu et al. (2018) and **Swan (2015)** focused on blockchain's role in authenticating and managing IoT devices. Their research showed that blockchain can be used to securely authenticate devices and ensure proper communication in healthcare IoT systems, eliminating central points of failure and enhancing device trust.

Interoperability is another key challenge, addressed by **Tremblay et al. (2017)** and **Zhang et al. (2019)**, who proposed that blockchain could enable secure, real-time communication between IoT devices, regardless of manufacturer, thereby improving coordination and reducing data fragmentation in healthcare systems.

Blockchain's potential in real-time monitoring and automation was explored by **Khan et al. (2018)** and **Patel et al. (2018)**. They highlighted how blockchain ensures the integrity of data from IoT devices, facilitating real-time alerts and automated actions, such as notifying healthcare providers of abnormal readings.

Additionally, blockchain can help ensure **regulatory compliance** and **data privacy**, as shown by **Petersen et al. (2019)**, who discussed blockchain's transparent audit trail for proving compliance with regulations like HIPAA and GDPR, and **Lemieux (2016)**, who emphasized how blockchain gives patients more control over their health data.

This review highlights the blockchain can significantly improve the management and security of IoT devices in healthcare, though challenges such as scalability and integration remain to be fully addressed.

2.2 DEFINITION OF PROBLEM STATEMENT

The primary goal of this project is healthcare industry is increasingly adopting Internet of Things (IoT) devices to monitor patient health, collect critical data, and improve the efficiency of care delivery. These devices, ranging from wearables to diagnostic equipment, generate vast amounts of sensitive data. However, managing and securing this data, ensuring interoperability between devices, and maintaining patient privacy pose significant challenges. Traditional centralized systems for managing IoT devices are vulnerable to data breaches, unauthorized access, and lack of transparency. Moreover, the lack of a standardized and decentralized framework complicates device authentication, communication, and data sharing, leading to potential security risks and inefficiencies.

To address these issues, there is a need for a **robust and secure management system** that can authenticate IoT devices, ensure real-time secure data transmission, guarantee data integrity, and improve device interoperability while adhering to healthcare privacy regulations.

Blockchain technology, with its **decentralized, immutable ledger** and cryptographic features, has the potential to provide an effective solution. However, its integration with IoT devices in healthcare requires overcoming challenges related to scalability, system integration, and regulatory compliance.

2.3 EXISTING SYSTEM

Healthcare systems primarily rely on centralized databases to store and manage the data generated by IoT devices. While these systems allow for easy access to data, they come with significant drawbacks. Centralized systems are vulnerable to cyberattacks, making them targets for hackers who can manipulate or steal sensitive patient data. Additionally, these systems create a single point of failure—if the central server experiences issues or downtime, access to crucial medical data and IoT devices is disrupted. Furthermore, centralized systems often lack transparency, meaning patients and healthcare providers may not have visibility into who is accessing the data or how it is being used, raising concerns about data privacy.

Another issue is the use of proprietary IoT devices from different manufacturers, which often have incompatible communication protocols and data formats. This results in interoperability problems between devices, making it difficult for healthcare systems to integrate devices into a cohesive network. The lack of standardization leads to data fragmentation, where patient data is scattered across different devices and platforms, hindering effective care coordination. Additionally, many IoT devices lack robust security measures, exposing them to the risk of unauthorized access or tampering.

Although blockchain technology has been explored for healthcare applications, most existing blockchain-based solutions focus on storing patient records and facilitating data sharing. These systems are often limited by scalability issues, especially when managing large amounts of real-time data generated by IoT devices. Blockchain also faces challenges in fully integrating with IoT systems, particularly in ensuring secure, real-time communication between devices. Moreover, many blockchain solutions in healthcare struggle to meet regulatory compliance standards such as HIPAA and GDPR, particularly regarding patient consent and data access control.

In terms of security, existing systems rely on encryption and access control to protect data. While these methods ensure data confidentiality, they do not fully guarantee data integrity or prevent unauthorized data manipulation. Additionally, these systems lack transparency regarding who accesses the data and when, which is crucial for maintaining privacy and security in healthcare.

Limitations of Existing System

- Scalability Issues: Blockchain networks can face challenges in scalability, particularly when handling large volumes of data generated by numerous IoT devices. As the number of devices and transactions increases, the blockchain may experience slow processing times and higher transaction costs, which can hinder real time data management essential in medical applications.
- Interoperability Challenges: Integrating blockchain solutions with existing healthcare systems and IoT devices can be complex. Different devices may use varying protocols and data formats, making it difficult to create a seamless interaction between blockchain and traditional systems. This lack of standardization can impede effective data exchange and usability.
- Data Privacy Concerns: While blockchain is known for its security and immutability, ensuring patient data privacy can be problematic. Sensitive medical information stored on a blockchain, even in encrypted form, may be at risk of exposure. Balancing transparency with the need for privacy in healthcare data is a significant challenge.
- Energy Consumption: Many blockchain networks, particularly those using proof of work consensus mechanisms, can be energy intensive. In the context of healthcare IoT, this high energy consumption can be a drawback, especially if devices are battery operated and require efficient energy management.
- Regulatory and Compliance Issues: The use of blockchain in healthcare is still evolving, and there may be regulatory uncertainties surrounding its implementation. Compliance with laws like HIPAA (Health Insurance Portability and Accountability Act) in the U.S. can be complex, as organizations must ensure that their blockchain solutions meet all legal requirements regarding data security and patient confidentiality

2.4 PROPOSED SYSTEM

The proposed system for managing Internet of Things (IoT) devices in the medical industry through blockchain technology aims to enhance security, interoperability, and data integrity. This system utilizes a decentralized blockchain network to securely store and share data generated by various IoT devices, such as wearable health monitors, smart medical devices, and diagnostic tools. Each device is assigned a unique digital identity on the blockchain, enabling secure authentication and access control. Data collected from these devices is encrypted and recorded in immutable blocks, ensuring that patient information remains tamper proof and can only be accessed by authorized personnel.

Smart contracts automate processes such as device registration, data sharing, and compliance with regulatory standards, thereby streamlining operations while reducing administrative burdens. Additionally, the system fosters seamless interoperability among diverse medical devices, allowing healthcare providers to gain a holistic view of patient data in real time. Overall, this blockchain based approach not only enhances the security and reliability of IoT devices in healthcare but also promotes patient privacy, data ownership, and trust among stakeholders in the medical ecosystem.

The proposed system for implementing blockchain technology in the management of Internet of Things (IoT) devices within the medical industry aims to enhance security, interoperability, and data integrity. This system utilizes a decentralized blockchain network to securely store and share data collected from various IoT devices, such as wearable health monitors, smart medical devices, and remote patient monitoring systems. Each IoT device is registered on the blockchain, creating a tamper proof record of its identity and data. The use of smart contracts automates workflows, such as data access permissions and compliance checks, ensuring that only authorized personnel can access sensitive patient information. This system facilitates real time data sharing among healthcare providers, patients, and researchers while maintaining strict adherence to privacy regulations like HIPAA.

The transparent and immutable nature of blockchain enhances accountability, enabling better tracking of device performance and patient outcomes. Overall, this blockchain based approach fosters a secure and efficient ecosystem for managing IoT devices in healthcare, ultimately improving patient care and operational efficiency.

Advantages of the Proposed System:

1. Enhanced Security and Data Integrity: Blockchain provides a decentralized and tamper proof ledger, ensuring that data generated by IoT devices is secure and cannot be altered without consensus. This protects sensitive medical information from unauthorized access and data breaches, which is crucial in healthcare.
2. Improved Interoperability: Blockchain can facilitate seamless communication between different IoT devices and healthcare systems, enabling better integration and interoperability. This ensures that data from various sources (like wearable devices, smart medical equipment, etc.) can be easily accessed and utilized by healthcare providers for better patient care.
3. Real Time Data Sharing: With blockchain, real time data from IoT devices can be shared securely among stakeholders (patients, healthcare providers, insurers, etc.) without delays. This immediacy is vital for timely decision making and can improve patient outcomes by allowing quicker responses to health changes.
4. Auditability and Traceability: Blockchain's immutable nature allows for complete audit trails of all transactions and data exchanges involving IoT devices. This traceability can enhance accountability, facilitate compliance with regulations, and provide insights into device performance and patient interactions.
5. Smart Contracts for Automation: Smart contracts can automate processes related to IoT devices, such as triggering alerts for anomalies, managing device maintenance schedules, or automatically processing payments for services rendered. This reduces manual intervention, minimizes errors, and increases operational efficiency in the healthcare sector.

2.5 OBJECTIVES

- **Enhance Data Security and Privacy**:

Ensure secure storage and transmission of sensitive medical data using blockchain encryption and immutability.

- **Enable Interoperability of IoT Devices:**

Facilitate seamless communication between IoT devices from different manufacturers through a decentralized blockchain framework.

- **Authenticate and Manage IoT Devices:**

Use blockchain to authenticate and securely register IoT devices in the healthcare system.

- **Real-Time Monitoring and Data Integrity:**

Ensure real-time monitoring and tamper-proof data collection from IoT devices with blockchain’s immutable ledger.

- **Automate Healthcare Workflows Using Smart Contracts:**

Integrate smart contracts to automate healthcare processes based on IoT device data and conditions.

- **Improve Regulatory Compliance and Transparency:**

Create an immutable audit trail to ensure compliance with regulations and enhance data transparency.

- **Scalability and Efficiency:**

Develop a scalable blockchain system capable of handling large volumes of real-time IoT data in healthcare.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

• Processor	:	Intel Core i5
• Hard disk	:	20GB.
• RAM	:	4GB.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

• Operating system	:	Windows 10
• Language	:	Python
• Back-End	:	Django-ORM
• Frame Work	:	Tkinter

3. SYSTEM ARCHITECTURE & DESIGN

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.

3.1 PROJECT ARCHITECTURE

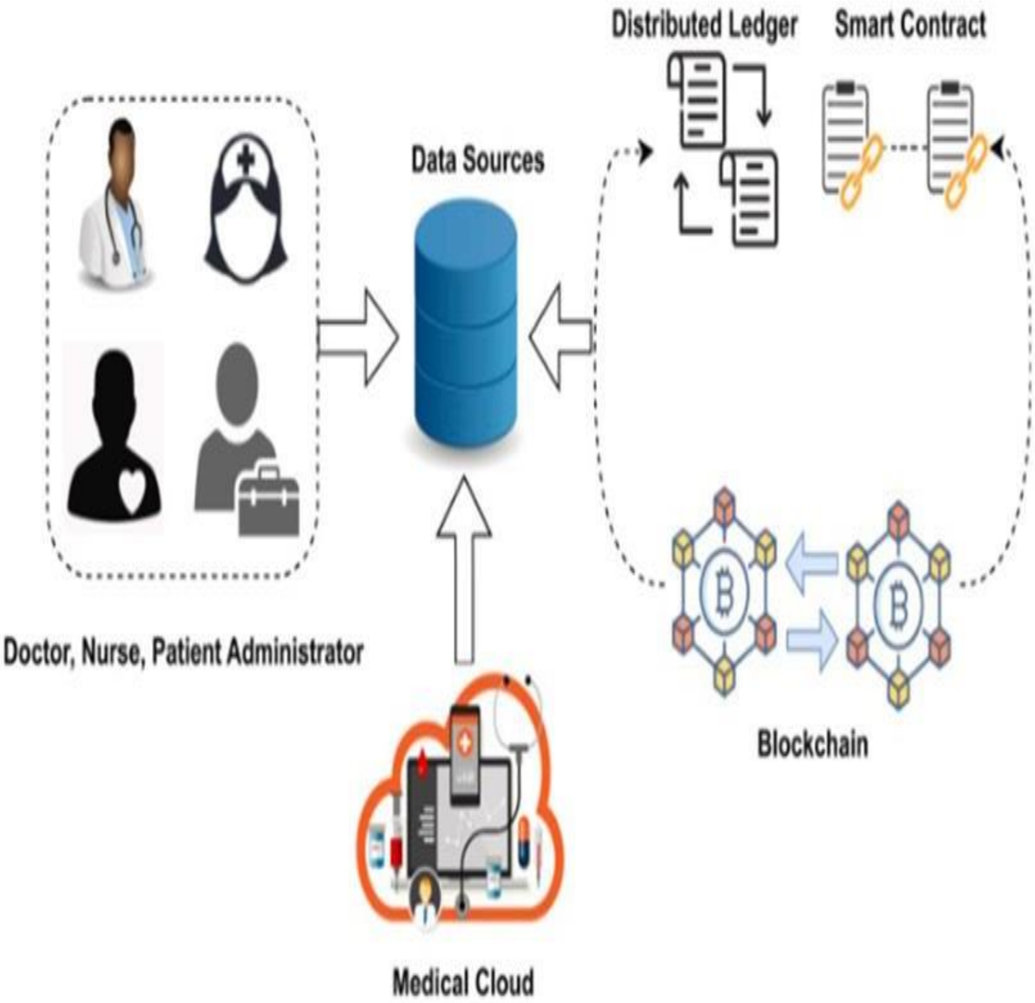


Figure 3.1: Project Architecture of A Deep Learning-Based Approach for Inappropriate Content Detection and Classification of YouTube Videos

3.2 DESCRIPTION

1. Data Sources (Medical Professionals and Administrators)

- The system gathers medical data from doctors, nurses, and patient administrators.
- These professionals input patient records, prescriptions, and medical histories into the system.

2. Medical Cloud (Storage and Processing)

- The medical cloud acts as an intermediary that stores and processes medical records before they are added to the blockchain.
- This allows for fast access and efficient management of healthcare data.

3. Blockchain Network

- A blockchain network ensures that medical data is secure, immutable, and decentralized.
- Patient records stored in the blockchain cannot be altered, ensuring trust and reliability in the system.

4. Distributed Ledger (Shared and Secure Data)

- A distributed ledger is used to maintain medical records across multiple nodes.
- This ensures that data is tamper-proof and accessible only to authorized parties.

5. Smart Contracts (Automated Rules and Security)

- Smart contracts enforce predefined rules, such as granting data access to healthcare providers or processing insurance claims automatically.
- This reduces manual work, increases efficiency, and ensures compliance with medical regulations.

3.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data flows within a system, showcasing its processes, data stores, and external entities. It is a vital tool in system analysis and design, helping stakeholders visualize the movement of information, identify inefficiencies, and optimize workflows.

In the proposed system, data flow begins at the hospital sensors, which continuously collect real-time patient health information such as heart rate, blood pressure, and oxygen levels. This raw data is transmitted securely to the medical cloud, which acts as a temporary storage and processing layer. The cloud formats, encrypts, and organizes the sensor data before sending it to the smart contract.

The smart contract plays a crucial role in validating the data and checking whether the sender and receiver have the correct permissions based on predefined rules. Once validated, the data is forwarded to the blockchain, where it is permanently stored as a secure, tamper-proof transaction. Each piece of information stored in the blockchain is linked in a cryptographic chain and made accessible through a distributed ledger. This ledger ensures that every stakeholder (like hospitals, regulatory bodies, and medical researchers) has access to a synchronized and immutable copy of the records. Meanwhile, users such as doctors, nurses, administrators, or even patients can request access to the data through the smart contract. If access permissions are verified, the user receives the requested data; otherwise, the request is denied. This flow ensures secure, decentralized, and transparent handling of sensitive medical data in the IoT-enabled healthcare environment.

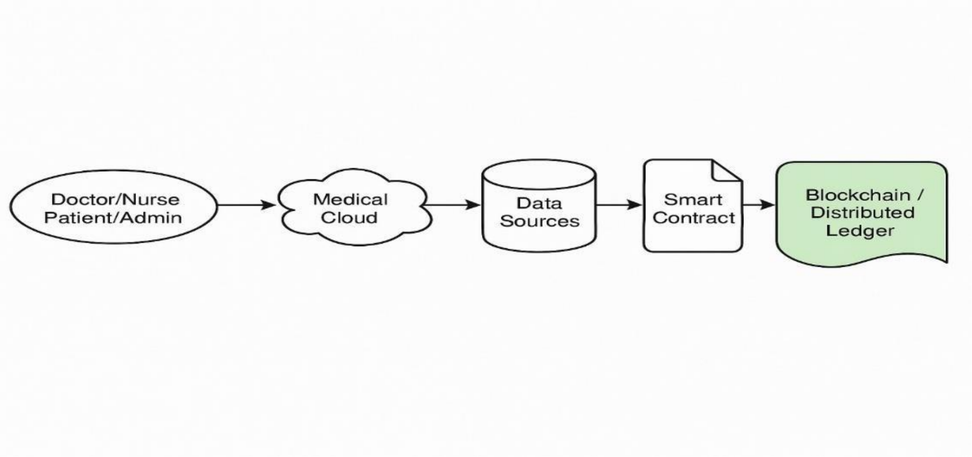


Figure 3.2: Dataflow Diagram of Blockchain for the management of internet of things devices in the medical industry

4. IMPLEMENTATION

The implementation phase of a project involves executing the planned strategies and tasks. It requires meticulous coordination, resource allocation, and monitoring to ensure that objectives are met efficiently. Effective implementation is crucial for achieving project goals and delivering expected outcomes within the set timeline and budget constraints.

4.1 ALGORITHMS

Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely used to secure sensitive data. Established by the National Institute of Standards and Technology (NIST) in 2001, AES replaced the older Data Encryption Standard (DES) due to its superior security and efficiency. AES encrypts data in fixed-size blocks of 128 bits and supports key lengths of 128, 192, or 256 bits, making it highly adaptable to various security needs. The encryption process involves multiple rounds of transformations, including SubBytes (substitution), ShiftRows (row shifting), MixColumns (column mixing), and AddRoundKey (XOR with a key schedule). Depending on the key size, AES performs 10, 12, or 14 rounds of encryption, ensuring robust data protection.

As a symmetric cipher, AES uses the same key for encryption and decryption, making it efficient for secure communication. It is widely implemented in HTTPS, TLS, Wi-Fi security (WPA2/WPA3), file encryption tools (BitLocker, VeraCrypt), and government data protection. The AES-256 variant is even approved for encrypting top-secret government information. Its resistance to brute-force attacks, differential cryptanalysis, and other cryptographic vulnerabilities makes AES one of the most secure encryption standards today. Additionally, its efficiency in both hardware and software implementations has made it the global standard for encrypting digital data.

Advantages of AES:

- **High Security** : AES provides strong encryption, making it resistant to brute-force attacks and cryptographic analysis. The large key sizes (128, 192, and 256 bits) ensure robust security.

- **Fast and Efficient**: AES is optimized for both hardware and software implementations, allowing quick encryption and decryption with minimal performance impact.

- **Widely Adopted :**AES is the global encryption standard, used in HTTPS, TLS, Wi-Fi security (WPA2/WPA3), secure messaging, and file encryption.

- **Low Memory Usage** :Unlike some other encryption algorithms, AES has low memory and processing power requirements, making it ideal for embedded systems and IoT devices.

Disadvantages of AES:

- Key Management Complexity :Since AES uses the same key for encryption and decryption (symmetric encryption), securely sharing and managing the key can be challenging.

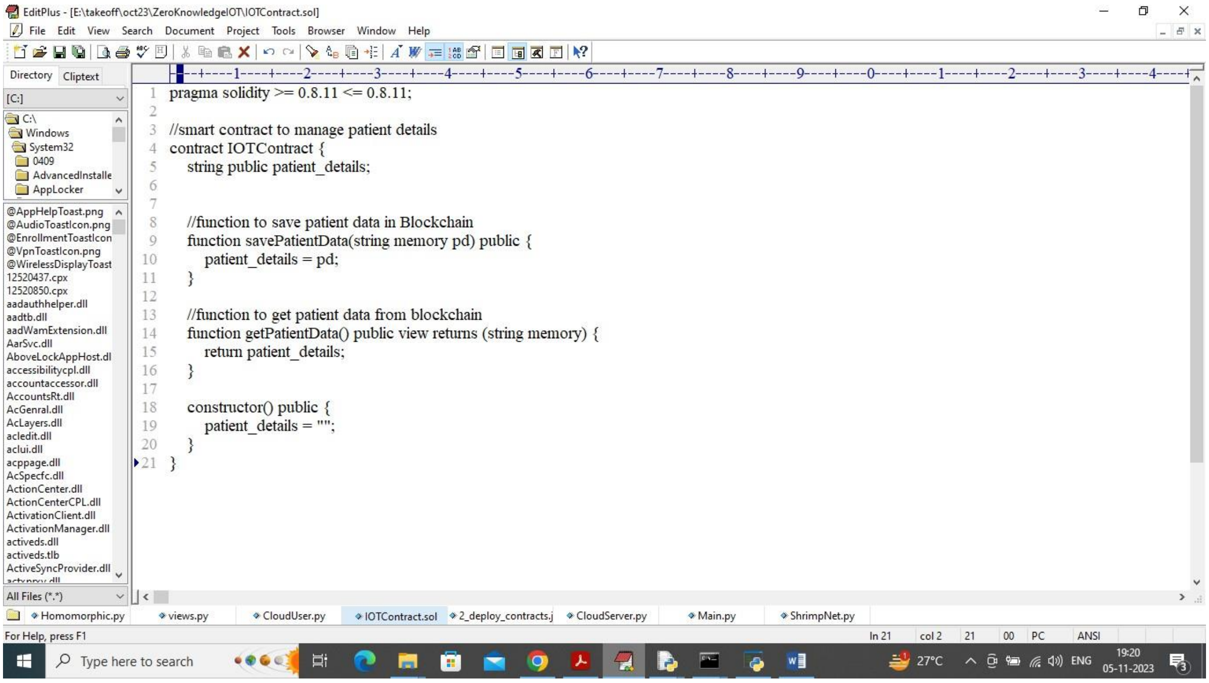
- Vulnerability to Side-Channel Attacks : Although AES is mathematically strong, poor implementation in hardware or software can make it vulnerable to side-channel attacks such as timing or power analysis attacks.

- Fixed Block Size : AES encrypts data in 128-bit blocks, which may require padding for data that doesn’t fit exactly, increasing overhead.

- Slower Than Some Stream Ciphers :For real-time applications like video streaming or voice communication, stream ciphers (e.g., ChaCha20) may perform better than AES in certain cases.

To implement this project we have generate simulation based IOT sensors and then equipped each sensor with ECC and Zero Knowledge private and public keys. This keys and Blockchain details will be stored in each sensor by Manufacturer and validated by certificate authorities or Node Verifier but we don’t have real IOT devices so we are adding keys to simulated IOT nodes.

To store IOT data in Blockchain we need to develop Smart Contract Code which will contains function to store and get data from Blockchain and in below screen we are showing Smart Contract code which is designed solidity programming.

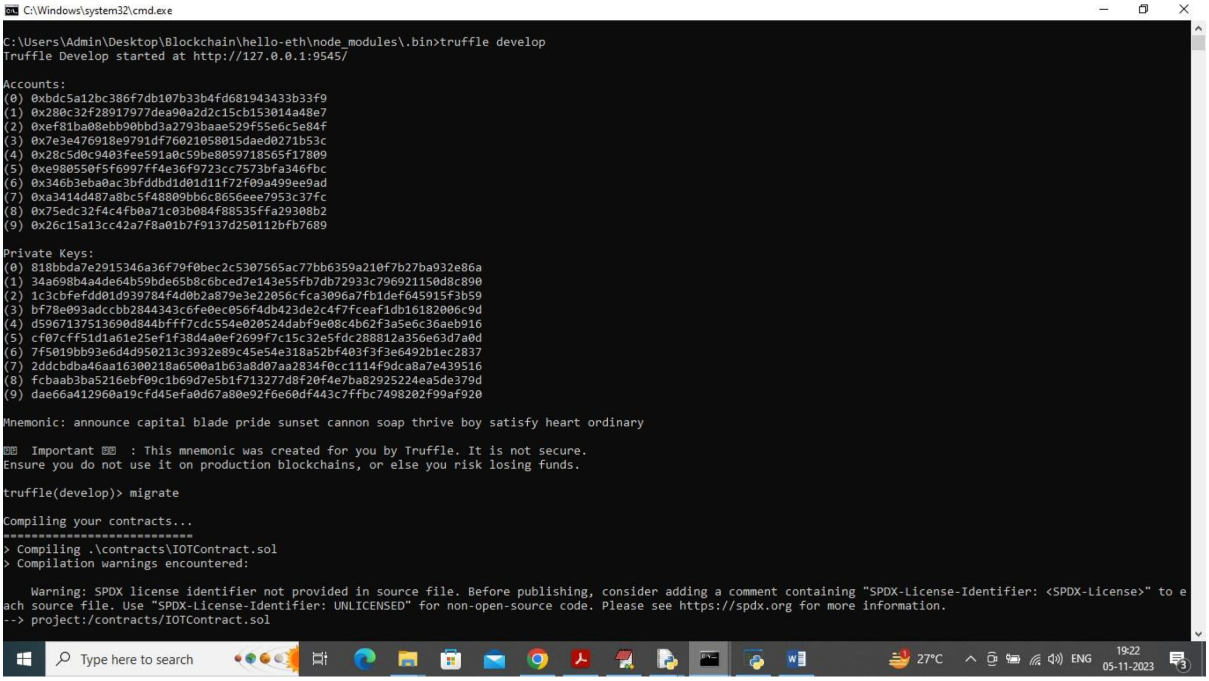


EditPlus - [E:\takeoff\oct23\ZeroKnowledgeIOT\IOTContract.sol]

```
pragma solidity >= 0.8.11 <= 0.8.11;

//smart contract to manage patient details
contract IOTContract {
    string public patient_details;


    //function to save patient data in Blockchain
    function savePatientData(string memory pd) public {
        patient_details = pd;
    }

    //function to get patient data from blockchain
    function getPatientData() public view returns (string memory) {
        return patient_details;
    }

    constructor() public {
        patient_details = "";
    }
}
```

In above screen smart contract code we defined function to save and get patient data and now we need to deploy this Smart Contract in Ethereum tool by following below steps

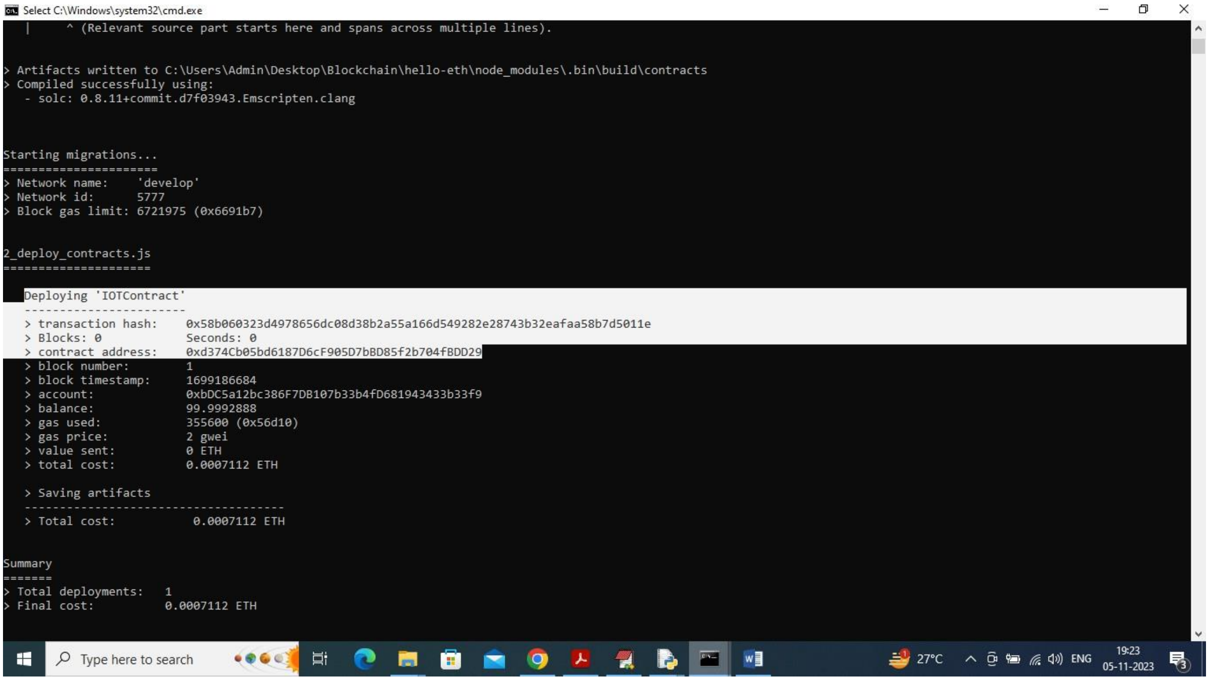
1) Go inside ‘hello-eth/node-modules/bin’ folder and then find and double click on ‘runBlockchain.bat’ file to start Ethereum tool and get below output



C:\Windows\system32\cmd.exe

```
C:\Users\Admin\Desktop\Blockchain\hello-eth\node_modules\.bin>truffle develop
Truffle Develop started at http://127.0.0.1:9545/

Accounts:
(0) 0xbdc5a12bc386f7db107b33b4fd681943433b33f9
(1) 0x280c32f28917977dea90a2d2c15cb153014a48e7
(2) 0xef81ba08ebb90bbd3a2793baae529f55e6c5e84f
(3) 0x7e3e476918e9791df76021058015daed0271b53c
(4) 0x28c5d0c9403fee591a0c59be8059718565f17809
(5) 0xe980550f5f6997ff4e36f9723cc7573bfa346fbc
(6) 0x346b3eba0ac3bfddbd1d01d11f72f09a499ee9ad
(7) 0xa3414d487a8bc5f48809bb6c8656eee7953c37fc
(8) 0x75edc32f4c4fb0a71c03b084f88535ffa29308b2
(9) 0x26c15a13cc42a7f8a01b7f9137d250112bfb7689

Private Keys:
(0) 818bbda7e2915346a36f79f0bec2c5307565ac77bb6359a210f7b27ba932e86a
(1) 34a698b4a4de64b59bde65b8c6bced7e143e55fb7db72933c796921150d8c890
(2) 1c3cbfefdd01d939784f4d0b2a879e3e22056cfca3096a7fb1def645915f3b59
(3) bf78e093adccbb2844343c6fe0ec056f4db423de2c4f7fceaf1db16182006c9d
(4) d5967137513690d844bfff7cdc554e020524dabf9e08c4b62f3a5e6c36aeb916
(5) cf07cff51d1a61e25ef1f38d4a0ef2699f7c15c32e5fdc288812a356e63d7a0d
(6) 7f5019bb93e6d4d950213c3932e89c45e54e318a52bf403f3f3e6492b1ec2837
(7) 2ddcbdba46aa16300218a6500a1b63a8d07aa2834f0cc1114f9dca8a7e439516
(8) fcbaab3ba5216ebf09c1b69d7e5b1f713277d8f20f4e7ba82925224ea5de379d
(9) dae66a412960a19cfd45efa0d67a80e92f6e60df443c7ffbc7498202f99af920

Mnemonic: announce capital blade pride sunset cannon soap thrive boy satisfy heart ordinary

⚠️  Important ⚠️  : This mnemonic was created for you by Truffle. It is not secure.
Ensure you do not use it on production blockchains, or else you risk losing funds.

truffle(develop)> migrate

Compiling your contracts...
===========================
> Compiling .\contracts\IOTContract.sol
> Compilation warnings encountered:

    Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to e
ach source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source code. Please see https://spdx.org for more information.
--> project:/contracts/IOTContract.sol
```

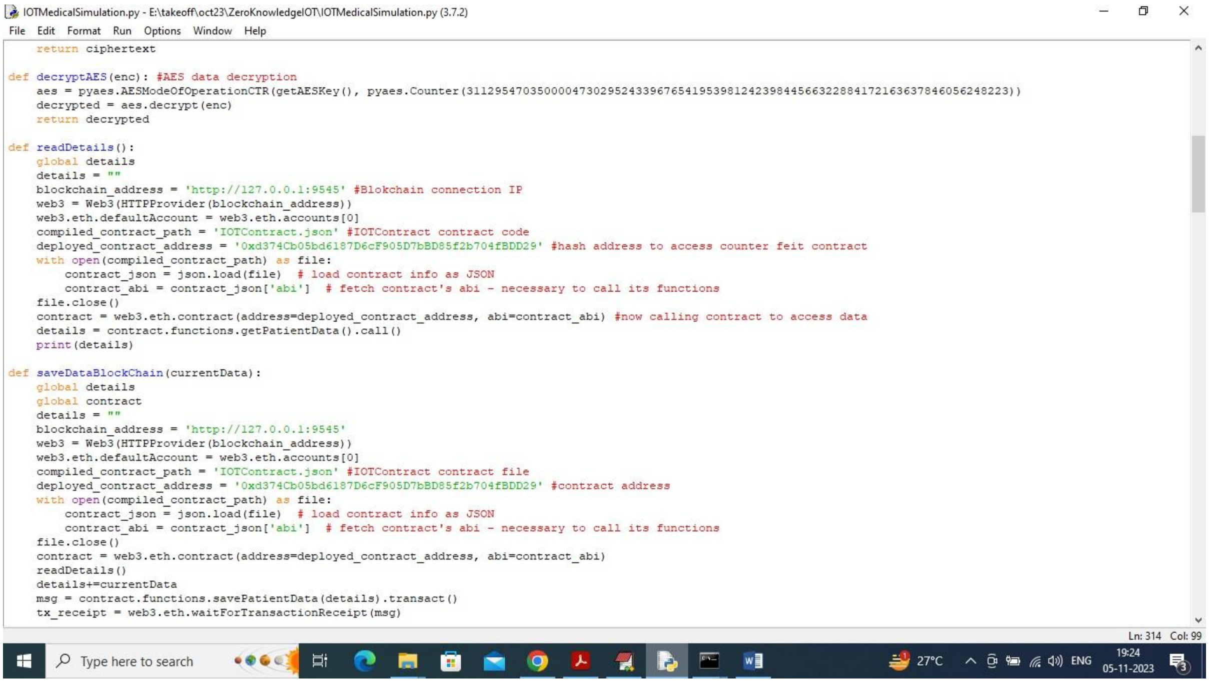
2) In above screen Ethereum started with default account address and private keys and now type command as ‘migrate’ and press enter key to deploy contract and get below output



Select C:\Windows\system32\cmd.exe

```
   |     ^ (Relevant source part starts here and spans across multiple lines).


> Artifacts written to C:\Users\Admin\Desktop\Blockchain\hello-eth\node_modules\.bin\build\contracts
> Compiled successfully using:
   - solc: 0.8.11+commit.d7f03943.Emscripten.clang



Starting migrations...
======================
> Network name:    'develop'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
=====================

   Deploying 'IOTContract'
   -----------------------
   > transaction hash:    0x58b060323d4978656dc08d38b2a55a166d549282e28743b32eafaa58b7d5011e
   > Blocks: 0            Seconds: 0
   > contract address:    0xd374Cb05bd6187D6cF905D7bBD85f2b704fBDD29
   > block number:        1
   > block timestamp:     1699186684
   > account:             0xbDC5a12bc386F7DB107b33b4fD681943433b33f9
   > balance:             99.9992888
   > gas used:            355600 (0x56d10)
   > gas price:           2 gwei
   > value sent:          0 ETH
   > total cost:          0.0007112 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:           0.0007112 ETH


Summary
=======
> Total deployments:   1
> Final cost:          0.0007112 ETH
```

3) In above screen in white colour text can see ‘IOT Contract’ smart contract deployed and we got contract address also and this address we need to specify in python code to call that contract for storage and reading data from Blockchain. In below screen showing python code calling smart contract using address



IOTMedicalSimulation.py - E:\takeoff\oct23\ZeroKnowledgeIOT\IOTMedicalSimulation.py (3.7.2)

```
    return ciphertext

def decryptAES(enc): #AES data decryption
    aes = pyaes.AESModeOfOperationCTR(getAESKey(), pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163637846056248223))
    decrypted = aes.decrypt(enc)
    return decrypted

def readDetails():
    global details
    details = ""
    blockchain_address = 'http://127.0.0.1:9545' #Blokchain connection IP
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'IOTContract.json' #IOTContract contract code
    deployed_contract_address = '0xd374Cb05bd6187D6cF905D7bBD85f2b704fBDD29' #hash address to access counter feit contract
    with open(compiled_contract_path) as file:
        contract_json = json.load(file)  # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi) #now calling contract to access data
    details = contract.functions.getPatientData().call()
    print(details)

def saveDataBlockChain(currentData):
    global details
    global contract
    details = ""
    blockchain_address = 'http://127.0.0.1:9545'
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'IOTContract.json' #IOTContract contract file
    deployed_contract_address = '0xd374Cb05bd6187D6cF905D7bBD85f2b704fBDD29' #contract address
    with open(compiled_contract_path) as file:
        contract_json = json.load(file)  # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
    readDetails()
    details+=currentData
    msg = contract.functions.savePatientData(details).transact()
    tx_receipt = web3.eth.waitForTransactionReceipt(msg)
```

4) In above python code read red colour comments to know about smart contract calling from python.

So by using above details Blockchain is ready and to simulate IOT we have designed following modules

1) Generate Medical IOT Sensors: using this module we will generate IOT sensors
2) Generate Private & Public Keys: using this module will generate ECC and zero knowledge keys
3) Start Simulation: using this module we will select one IOT as patient sensor and the randomly generate BP and Heart rate and then encrypt details using AES and the generate ECC and ZKP signatures and then send packet to Hospital. Hospital will verify signatures and upon verified will save packet to Blockchain
4) Verification Time Chart: using this module we will plot ECC and ZKP verification time comparison chart
5) Read Data from Blockchain: using this module we will read all patients data stored in Blockchain and display to Hospital employees.

4.2 SAMPLE CODE

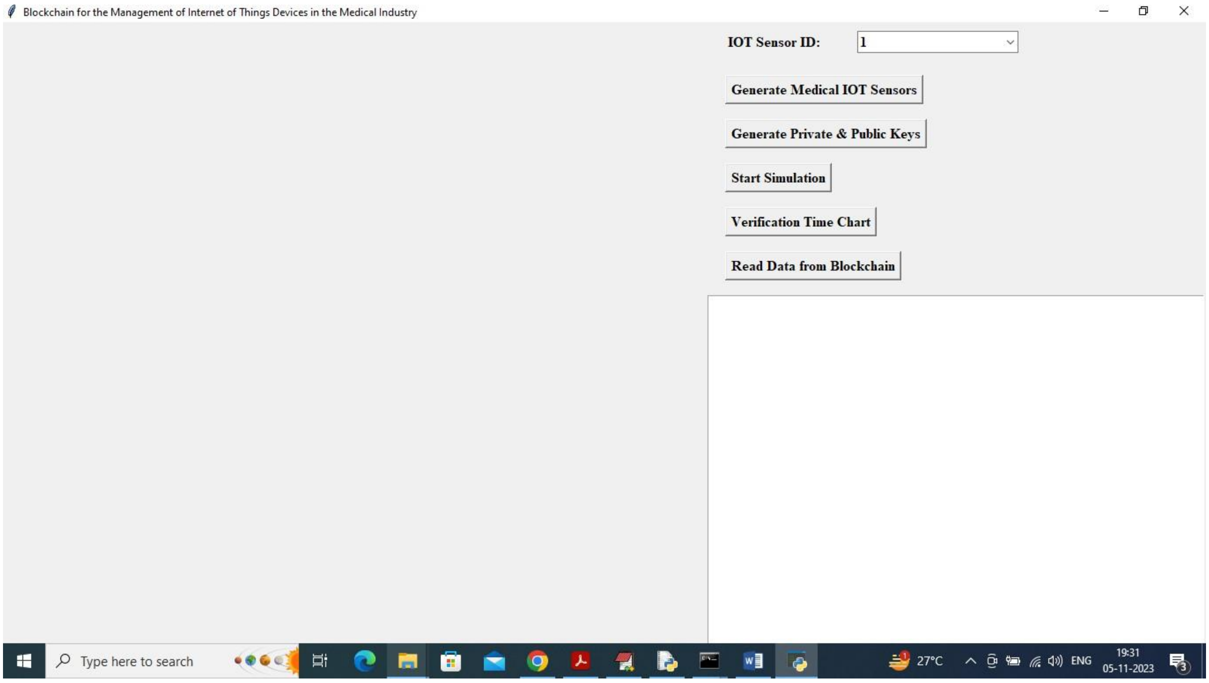
```
import tkinter
from tkinter import *
import math
import random
from threading import Thread
from collections import defaultdict
from tkinter import ttk
import matplotlib.pyplot as plt
import numpy as np
import time
import random
import numpy as np
import hmac
import hashlib
import base64
import random
import socket
import pickle
import json
from web3 import Web3, HTTPProvider
import timeit
import ECC
from ZeroKnowledge import ZKProof
from datetime import datetime
import timeit
import pyaes, pbkdf2, binascii, os, secrets
import webbrowser

global mobile
global labels
global mobile_x
global mobile_y
global text
global canvas
global source_list, dest_list, tf1
global filename
global p1,p2,p3
global line1,line2,line3
option = 0
global root
rewards = []
global details
ecc = []
zkp = []
global zkp_proof, details

def getAESKey(): #generating key with PBKDF2 for AES
    password = "s3cr3t*c0d3"
    passwordSalt = '76895'
    key = pbkdf2.PBKDF2(password, passwordSalt).read(32)
    return key

def encryptAES(plaintext): #AES data encryption
    aes = pyaes.AESModeOfOperationCTR(getAESKey(),
pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163
637846056248223))
    ciphertext = aes.encrypt(plaintext)
    return ciphertext

def decryptAES(enc): #AES data decryption
    aes = pyaes.AESModeOfOperationCTR(getAESKey(),
pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163
637846056248223))
    decrypted = aes.decrypt(enc)
    return decrypted

def readDetails():
    global details
    details = ""
    blockchain_address = 'http://127.0.0.1:9545' #Blokchain connection IP
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'IOTContract.json' #IOTContract contract code
    deployed_contract_address = '0x1889207f21FDe8284E0C9F4D056f80F36753CE67'
#hash address to access counter feit contract
    with open(compiled_contract_path) as file:
        contract_json = json.load(file)  # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi) #now
calling contract to access data
    details = contract.functions.getPatientData().call()
    print(details)

def saveDataBlockChain(currentData):
    global details
    global contract
    details = ""
    blockchain_address = 'http://127.0.0.1:9545'
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'IOTContract.json' #IOTContract contract file
    deployed_contract_address = '0x1889207f21FDe8284E0C9F4D056f80F36753CE67'
#contract address
    with open(compiled_contract_path) as file:
        contract_json = json.load(file)  # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
    readDetails()
    details+=currentData
    msg = contract.functions.savePatientData(details).transact()
    tx_receipt = web3.eth.waitForTransactionReceipt(msg)

def getDistance(iot_x,iot_y,x1,y1):
    flag = False
    for i in range(len(iot_x)):
        dist = math.sqrt((iot_x[i] - x1)*2 + (iot_y[i] - y1)*2)
        if dist < 80:
            flag = True
            break
    return flag


def startDataTransferSimulation(message, ecc_sign, zkp_sign, aes_data,
text,canvas,line1,line2,x1,y1,x2,y2,x3,y3):
    class SimulationThread(Thread):
        def _init_(self, message, ecc_sign, zkp_sign, aes_data,
text,canvas,line1,line2,x1,y1,x2,y2,x3,y3):
            Thread._init_(self)
            self.canvas = canvas
            self.line1 = line1
            self.line2 = line2
            self.x1 = x1
            self.y1 = y1
            self.x2 = x2
            self.y2 = y2
            self.x3 = x3
            self.y3 = y3
            self.text = text
            self.message  =  message
            self.ecc_sign  =  ecc_sign
            self.zkp_sign = zkp_sign
            self.aes_data = aes_data

        def run(self):
            global zkp_proof
            for i in range(0,3):
                self.canvas.delete(self.line1)
                self.canvas.delete(self.line2)
                time.sleep(1)
                self.line1 = canvas.create_line(self.x1, self.y1,self.x2, self.y2,fill='black',width=3)
                self.line2 = canvas.create_line(self.x2, self.y2,self.x3, self.y3,fill='black',width=3)
                time.sleep(1)
            self.canvas.delete(self.line1)
            self.canvas.delete(self.line2)
            canvas.update()
            ecc_verify = ECC.eccVerify(self.ecc_sign, self.message.encode())
            self.text.insert(END,"Generated ECC Sign: "+str(self.ecc_sign)+"\n")
            if ecc_verify:
                self.text.insert(END,"ECC Verification Successful\n")
            else:
                self.text.insert(END,"ECC Verification Failed\n")
            zkp_verify = zkp_proof.verify(message)
            self.text.insert(END,"Generated ZKP Sign: "+str(self.zkp_sign)+"\n")
            if ecc_verify:
                self.text.insert(END,"ZKP Verification Successful\n")
            else:
                self.text.insert(END,"ZKP Verification Failed\n")
            aes_decrypt = decryptAES(self.aes_data)
            text.insert(END,"Received & AES Decrypted Packet = "+str(aes_decrypt.decode()))

    newthread = SimulationThread(message, ecc_sign, zkp_sign,
aes_data,text,canvas,line1,line2,x1,y1,x2,y2,x3,y3)
    newthread.start()

def generateKeys():
    global zkp_proof
    text.delete('1.0', END)
    zkp_proof = ZKProof()
    ECC.generateECCKey()
    text.insert(END,"Private Keys : "+str(ECC.private_key)+"\n")
    text.insert(END,"Public Keys : "+str(ECC.public_key)+"\n\n")

def generateIOTNetwork():
    global mobile
    global labels
    global mobile_x
    global mobile_y
    global source_list, dest_list
    mobile = []
    mobile_x = []
    mobile_y = []
    labels = []
    canvas.update()

    x = 5
    y = 350
    mobile_x.append(x)
    mobile_y.append(y)
    name = canvas.create_oval(x,y,x+40,y+40, fill="blue")
    lbl = canvas.create_text(x+20,y-10,fill="darkblue",font="Times 7 italic
bold",text="Hospital")
    labels.append(lbl)
    mobile.append(name)
    rewards.append(0)
    for i in range(1,20):
        run = True
        while run == True:
            x = random.randint(100, 450)
            y = random.randint(50, 600)
            flag = getDistance(mobile_x,mobile_y,x,y)
            if flag == False:
                rewards.append(0)
                mobile_x.append(x)
                mobile_y.append(y)
                run = False
                name = canvas.create_oval(x,y,x+40,y+40, fill="red")
                lbl = canvas.create_text(x+20,y-10,fill="darkblue",font="Times 8 italic
bold",text="P"+str(i))
                labels.append(lbl)
                mobile.append(name)

def startSimulation():
    global option
    global line1,line2,line3, details
    global source_list, dest_list, ecc, zkp, zkp_proof
    text.delete('1.0', END)
    src = int(source_list.get())
    dest = 0
    start1 = timeit.default_timer()
    text.insert(END,"Selected Patient Sensor is : "+str(src)+"\n")
    if option == 1:
        canvas.delete(line1)
        canvas.delete(line2)
        canvas.update()
    src_x = mobile_x[src]
    src_y = mobile_y[src]
    des_x = mobile_x[dest]
    des_y = mobile_y[dest]
    distance = 10000
    distance1 = 10000
    hop = -1
    neighbours = []
    for i in range(1,20):
        temp_x = mobile_x[i]
        temp_y = mobile_y[i]
        if i != src and i != dest:
            dist1 = math.sqrt((src_x - temp_x)*2 + (src_y - temp_y)*2)
            if dist1 < distance:
                distance = dist1
                neighbours.append(i)
    for i in range(len(neighbours)):
        nei = neighbours[i]
        temp_x = mobile_x[nei]
        temp_y = mobile_y[nei]
        dist1 = math.sqrt((des_x - temp_x)*2 + (des_y - temp_y)*2)
        if dist1 < distance1:
            distance1 = dist1
            hop = nei
    if hop != -1:
        hop = hop + 1
        bp = random.randint(60, 150)
        heart = random.randint(30, 90)
        now = datetime.now()
        dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
        message = "P"+str(src)+" BP="+str(bp)+" Heart="+str(heart)+" "+dt_string
        start_time = timeit.default_timer()
        ecc_sign = ECC.eccSign(message.encode())
        aes_data = encryptAES(message)
        end_time = timeit.default_timer()
        ecc_time = end_time - start_time
        ecc.append(ecc_time)
        start_time = timeit.default_timer()
        zkp_sign = zkp_proof.generate_proof(message)
        end_time = timeit.default_timer()
        zkp_time = end_time - start_time
        zkp.append(zkp_time)
        readDetails()
        saveDataBlockChain(message+"\n")
        text.insert(END,"Sending Packet : "+message+"\n")
        text.insert(END,"ECC Signature : "+str(ecc_sign)+"\n")
        text.insert(END,"ZKP Signature : "+str(zkp_sign)+"\n")
        text.insert(END,"AES Encrypted Data : "+str(aes_data)+"\n\n")
        line1 = canvas.create_line(mobile_x[src]+20, mobile_y[src]+20,mobile_x[hop]+20,
mobile_y[hop]+20,fill='black',width=3)
        line2 = canvas.create_line(mobile_x[hop]+20, mobile_y[hop]+20,mobile_x[dest]+20,
mobile_y[dest]+20,fill='black',width=3)
        startDataTransferSimulation(message, ecc_sign, zkp_sign, aes_data,
text,canvas,line1,line2,(mobile_x[src]+20),(mobile_y[src]+20),(mobile_x[hop]+20),(mobile
_y[hop]+20),(mobile_x[dest]+20),(mobile_y[dest]+20))
        option = 1
    else:
        text.insert(END,"Unable to report data to Publisher. Try another participant\n")

def graph():
    global zkp, ecc
    plt.figure(figsize=(10,6))
    plt.grid(True)
    plt.xlabel('Number of Verifications')
    plt.ylabel('Running Time')
    plt.plot(ecc, 'ro-', color = 'green')
    plt.plot(zkp, 'ro-', color = 'blue')
    plt.legend(['ECC Verification', 'ZKP Verification'], loc='upper left')
    plt.title('Algorithms Running Time Graph')
    plt.show()
    z
def readPatientData():
    global details
    output = '<table border=1 align=center>'
    output+='<tr><th><font size=3 color=black>Patient ID</font></th>'
    output+='<th><font size=3 color=black>Blood Pressure</font></th>'
    output+='<th><font size=3 color=black>Heart Rate</font></th>'
    output+='<th><font size=3 color=black>Date</font></th>'
    output+='<th><font size=3 color=black>Time</font></th></tr>'
    readDetails()
    arr = details.split("\n")
    for i in range(len(arr)-1):
        values = arr[i].split(" ")
        output+='<tr><td><font size=3 color=black>'+values[0]+'</font></td>'
        output+='<td><font size=3 color=black>'+values[1]+'</font></td>'
        output+='<td><font size=3 color=black>'+values[2]+'</font></td>'
        output+='<td><font size=3 color=black>'+values[3]+'</font></td>'
        output+='<td><font size=3 color=black>'+values[4]+'</font></td></tr>'
    output += "</table><br/><br/><br/><br/>"
    f = open("output.html", "w")
    f.write(output)
    f.close()
    webbrowser.open("output.html",new=1)



def Main():
    global root
    global tf1
    global text
    global canvas
    global source_list, dest_list, tf1
    root = tkinter.Tk()
    root.geometry("1300x1200")
    root.title("Blockchain for the Management of Internet of Things Devices in the Medical
Industry")
    root.resizable(True,True)
    font1 = ('times', 12, 'bold')

    canvas = Canvas(root, width = 800, height = 700)
    canvas.pack()

    l1 = Label(root, text='IOT Sensor ID:')
    l1.config(font=font1)
    l1.place(x=820,y=10)

    mid = []
    for i in range(1,20):
        mid.append(str(i))
    source_list = ttk.Combobox(root,values=mid,postcommand=lambda:
source_list.configure(values=mid))
    source_list.place(x=970,y=10)
    source_list.current(0)
    source_list.config(font=font1)

    createButton = Button(root, text="Generate Medical IOT Sensors",
command=generateIOTNetwork)
    createButton.place(x=820,y=60)
    createButton.config(font=font1)

    keysButton = Button(root, text="Generate Private & Public Keys",
command=generateKeys)
    keysButton.place(x=820,y=110)
    keysButton.config(font=font1)

    startButton = Button(root, text="Start Simulation", command=startSimulation)
    startButton.place(x=820,y=160)
    startButton.config(font=font1)

    graphButton = Button(root, text="Verification Time Chart", command=graph)
    graphButton.place(x=820,y=210)
    graphButton.config(font=font1)

    readButton = Button(root, text="Read Data from Blockchain", command=readPatientData)
    readButton.place(x=820,y=260)
    readButton.config(font=font1)

    text=Text(root,height=25,width=70)
    scroll=Scrollbar(text)
    text.configure(yscrollcommand=scroll.set)
    text.place(x=800,y=310)


    root.mainloop()


if _name== 'main_' :
    Main ()
```

5. RESULTS & DISCUSSION

The following screenshots showcase the results of our project, highlighting key features and functionalities. These visual representations provide a clear overview of how the system performs under various conditions, demonstrating its effectiveness and user interface. The screenshots serve as a visual aid to support the project's technical and operational achievements.

5.1 Project Execution Screen

To run project double click on ‘run.bat’ file to get below screen



Blockchain for the Management of Internet of Things Devices in the Medical Industry
IOT Sensor ID:
1
Generate Medical IOT Sensors
Generate Private & Public Keys
Start Simulation
Verification Time Chart
Read Data from Blockchain
Type here to search
27°C
ENG
19:31
05-11-2023

Fig 5.1 Displaying the initial project interface after running run.bat.

In above screen click on ‘Generate Medical IOT Sensors’ button to generate IOT sensors and get below page

5.2 IoT Sensor Generation

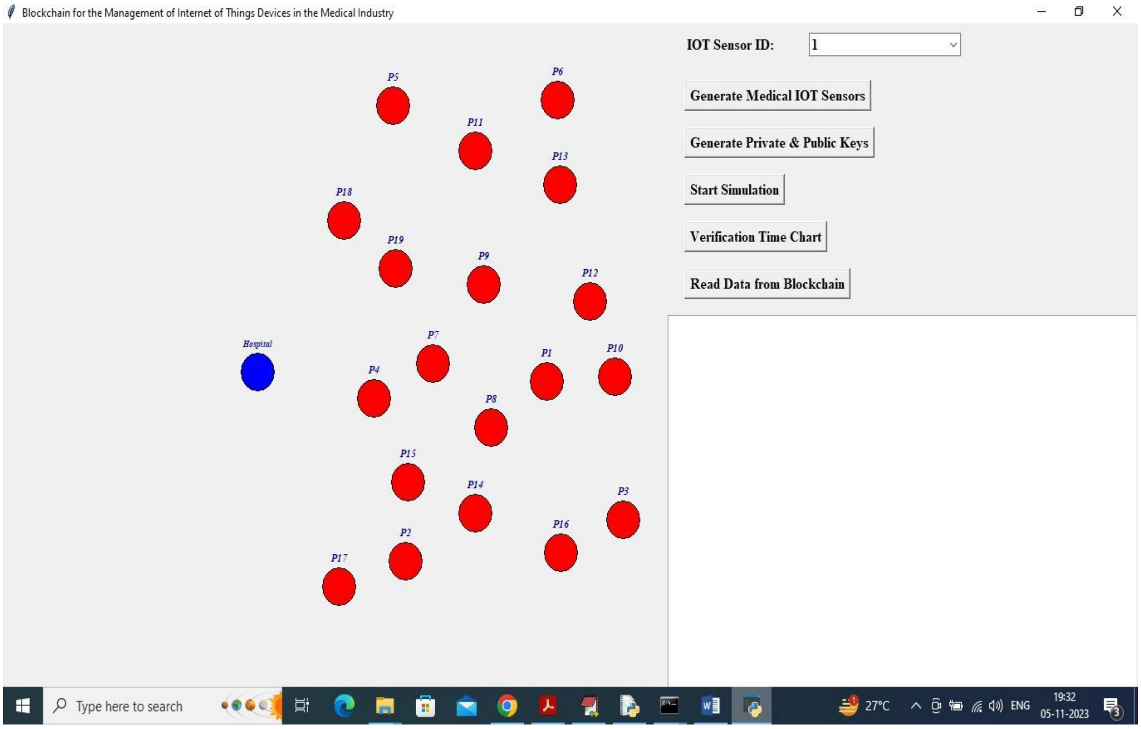
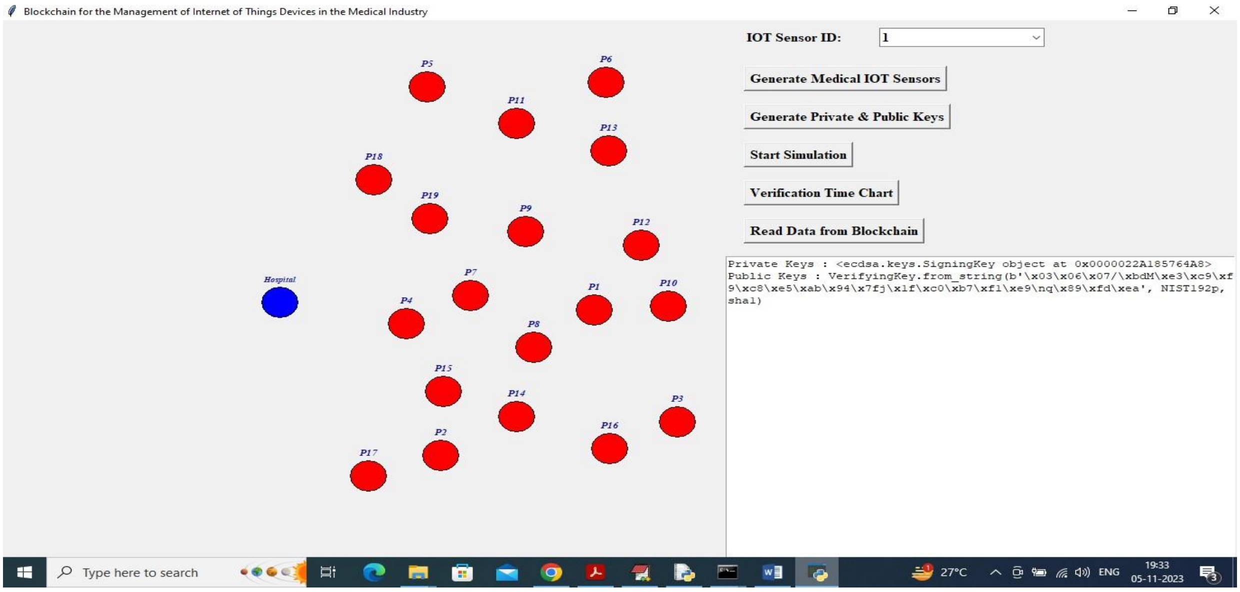

Blockchain for the Management of Internet of Things Devices in the Medical Industry
IOT Sensor ID:
1
Generate Medical IOT Sensors
Generate Private & Public Keys
Start Simulation
Verification Time Chart
Read Data from Blockchain
P5
P6
P11
P13
P18
P19
P9
P12
Hospital
P7
P4
P1
P10
P8
P15
P14
P3
P16
P2
P17
Type here to search
27°C
ENG
19:32
05-11-2023

Fig 5.2 IoT Sensor Generation

In above screen all red colour circles are the patients IOT sensors and blue colour circle is the Hospital Blockchain server and now click on 'Generate Private & Public Keys' button to generate keys and get below output.

5.3 IoT Sensor Network Visualization

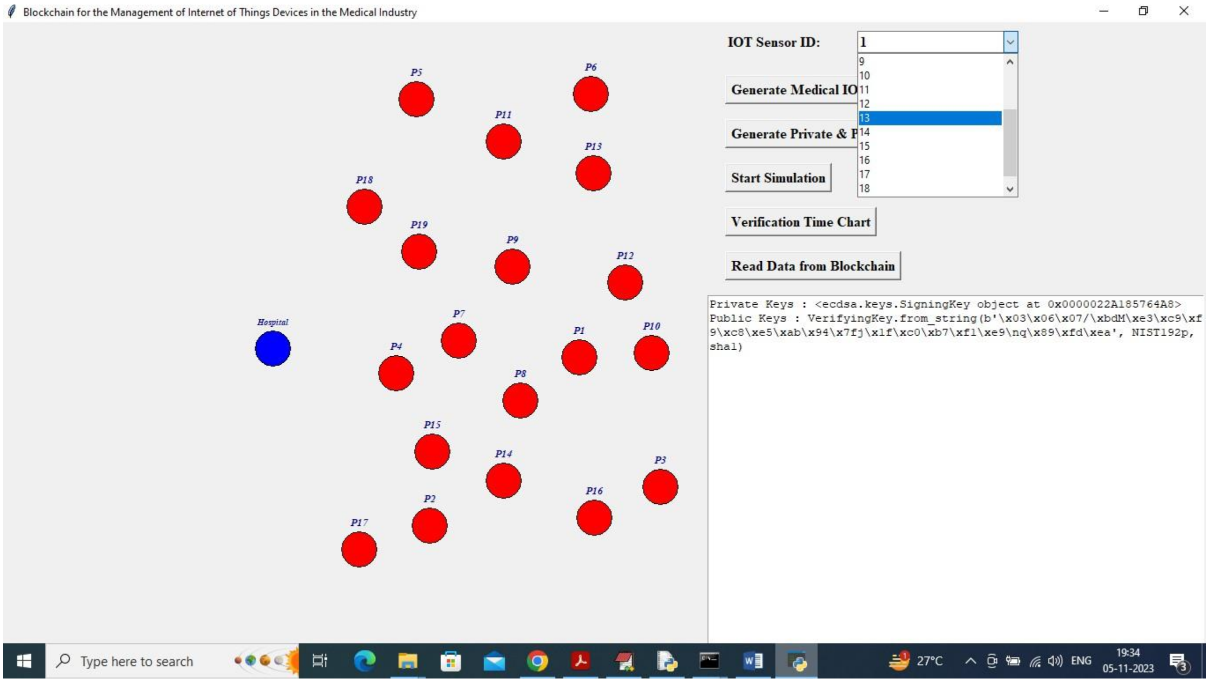

Blockchain for the Management of Internet of Things Devices in the Medical Industry
IOT Sensor ID:
1
Generate Medical IOT Sensors
Generate Private & Public Keys
Start Simulation
Verification Time Chart
Read Data from Blockchain
P5
P6
P11
P13
P18
P19
P9
P12
Hospital
P7
P4
P1
P10
P8
P15
P14
P3
P16
P2
P17
Type here to search
27°C
ENG
19:33
05-11-2023

```
Private Keys : <ecdsa.keys.SigningKey object at 0x0000022A185764A8>
Public Keys : VerifyingKey.from_string(b'\x03\x06\x07/\xbdM\xe3\xc9\xf
9\xc8\xe5\xab\x94\x7fj\x1f\xc0\xb7\xf1\xe9\nq\x89\xfd\xea', NIST192p,
sha1)
```

Fig 5.3 IoT Network Visualization

In above screen in text area we can see generated keys and now from first drop down box you can select some sensor ID and then click on ‘Start Simulation’ button to send packet to hospital node.

5.4 Key Generation

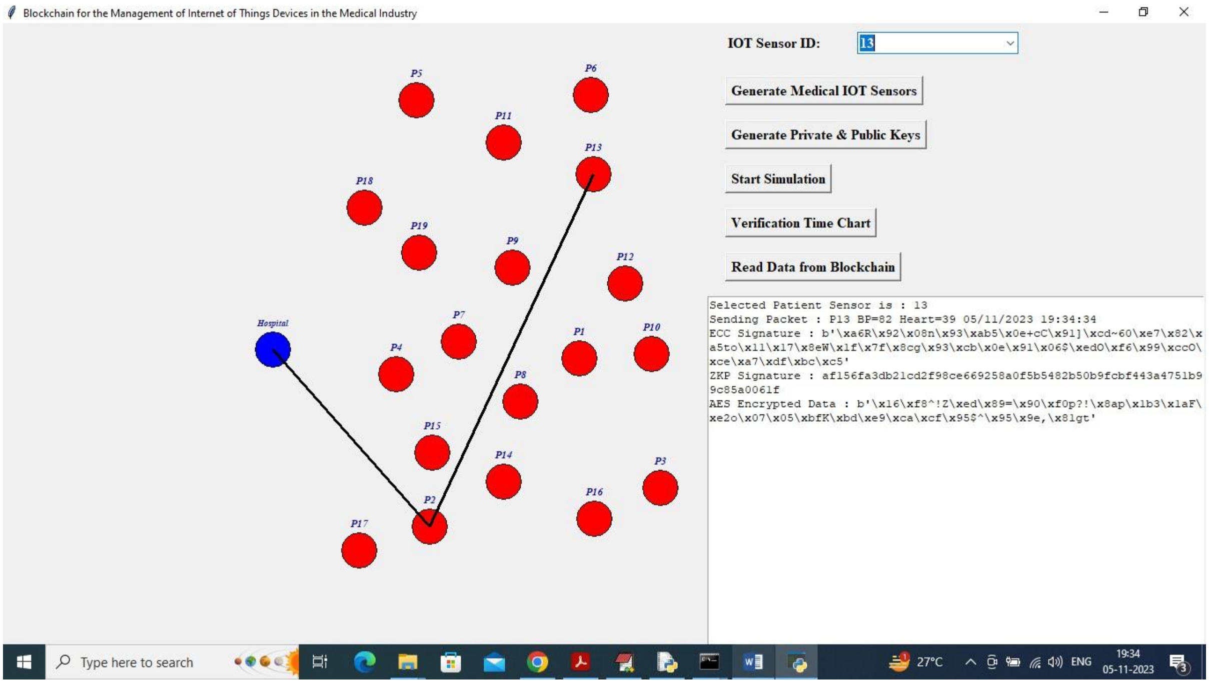


```
Private Keys : <ecdsa.keys.SigningKey object at 0x0000022A185764A8>
Public Keys : VerifyingKey.from_string(b'\x03\x06\x07/\xbdM\xe3\xc9\xf
9\xc8\xe5\xab\x94\x7fj\x1f\xc0\xb7\xf1\xe9\nq\x89\xfd\xea', NIST192p,
sha1)
```

Fig 5.4 Key Generation

In above screen selecting patient sensor as 13 and now click on ‘Start Simulation’ button to get below output

5.5 Packet Transmission Initialization


Blockchain for the Management of Internet of Things Devices in the Medical Industry
P5
P6
P11
P13
P18
P19
P9
P12
Hospital
P7
P4
P1
P10
P8
P15
P14
P3
P2
P16
P17
Generate Medical IOT Sensors
Generate Private & Public Keys
Start Simulation
Verification Time Chart
Read Data from Blockchain
Type here to search
27°C
ENG
19:34
05-11-2023

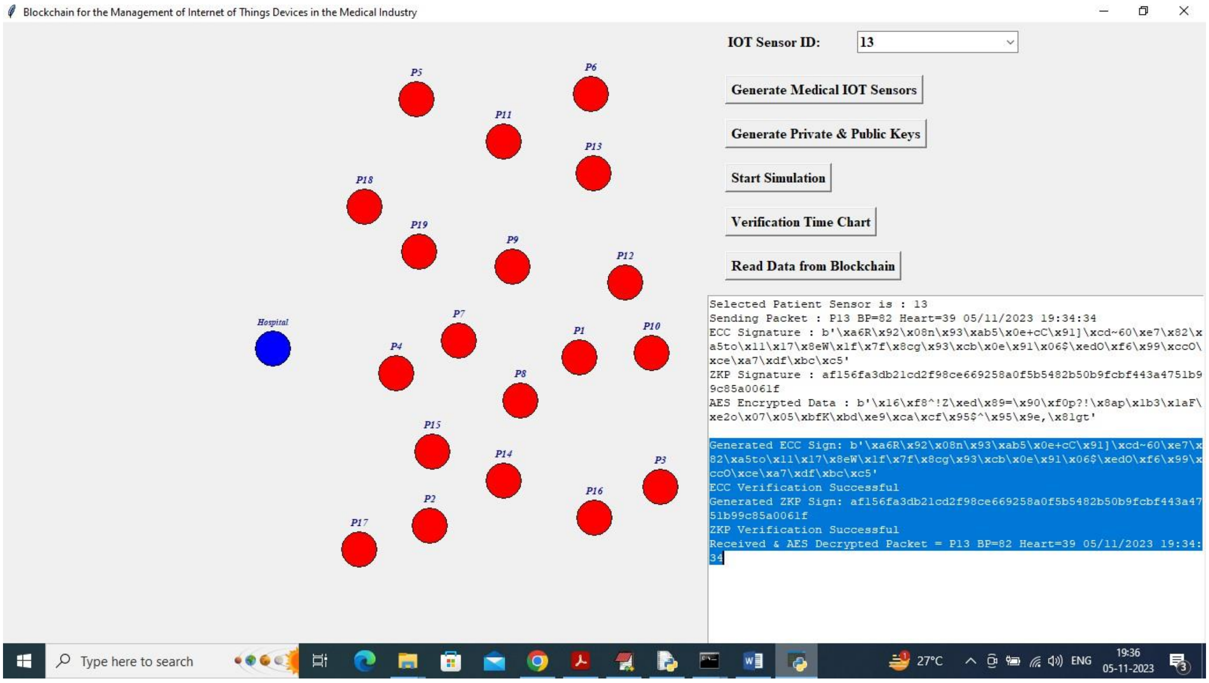
IOT Sensor ID: 13

```
Selected Patient Sensor is : 13
Sending Packet : P13 BP=82 Heart=39 05/11/2023 19:34:34
ECC Signature : b'\xa6R\x92\x08n\x93\xab5\x0e+cC\x91]\xcd~60\xe7\x82\x
a5to\x11\x17\x8eW\x1f\x7f\x8cg\x93\xcb\x0e\x91\x06$\xedO\xf6\x99\xccO\
xce\xa7\xdf\xbc\xc5'
ZKP Signature : af156fa3db21cd2f98ce669258a0f5b5482b50b9fcbf443a4751b9
9c85a0061f
AES Encrypted Data : b'\x16\xf8^!Z\xed\x89=\x90\xf0p?!\x8ap\x1b3\x1aF\
xe2o\x07\x05\xbfK\xbd\xe9\xca\xcf\x95$^\x95\x9e,\x81gt'
```

Fig 5.5 Packet Transmission Initialization

In above screen P13 is sending packet to sensor which is nearest and hospital and then that sensor will send to hospital. In text area we can see generated signatures and can see encrypted data. After receiving data hospital will perform verification and then will get below output

5.6 Packet Transmission Process


Blockchain for the Management of Internet of Things Devices in the Medical Industry
Generate Medical IOT Sensors
Generate Private & Public Keys
Start Simulation
Verification Time Chart
Read Data from Blockchain
P5
P6
P11
P13
P18
P19
P9
P12
Hospital
P7
P4
P1
P10
P8
P15
P14
P3
P16
P2
P17
Type here to search
27°C
ENG
19:36
05-11-2023

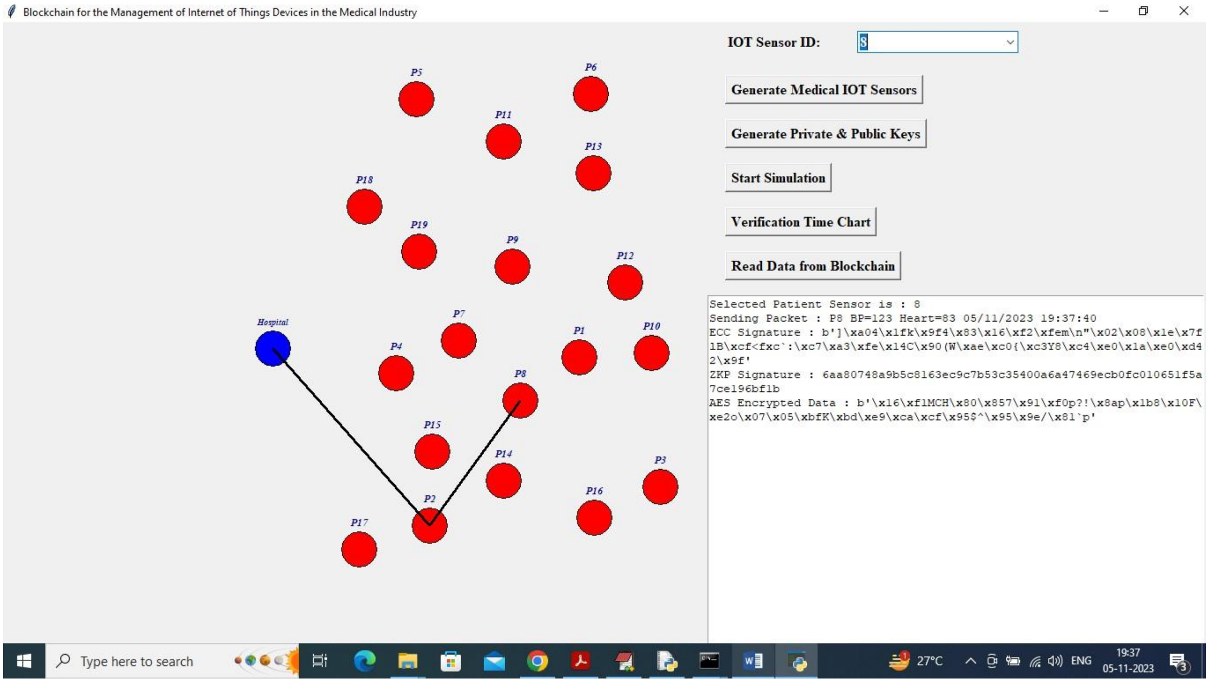
IOT Sensor ID: 13

```
Selected Patient Sensor is : 13
Sending Packet : P13 BP=82 Heart=39 05/11/2023 19:34:34
ECC Signature : b'\xa6R\x92\x08n\x93\xab5\x0e+cC\x91]\xcd~60\xe7\x82\x
a5to\x11\x17\x8eW\x1f\x7f\x8cg\x93\xcb\x0e\x91\x06$\xedO\xf6\x99\xccO\
xce\xa7\xdf\xbc\xc5'
ZKP Signature : af156fa3db21cd2f98ce669258a0f5b5482b50b9fcbf443a4751b9
9c85a0061f
AES Encrypted Data : b'\x16\xf8^!Z\xed\x89=\x90\xf0p?!\x8ap\x1b3\x1aF\
xe2o\x07\x05\xbfK\xbd\xe9\xca\xcf\x95$^\x95\x9e,\x81gt'

Generated ECC Sign: b'\xa6R\x92\x08n\x93\xab5\x0e+cC\x91]\xcd~60\xe7\x
82\xa5to\x11\x17\x8eW\x1f\x7f\x8cg\x93\xcb\x0e\x91\x06$\xedO\xf6\x99\x
ccO\xce\xa7\xdf\xbc\xc5'
ECC Verification Successful
Generated ZKP Sign: af156fa3db21cd2f98ce669258a0f5b5482b50b9fcbf443a47
51b99c85a0061f
ZKP Verification Successful
Received & AES Decrypted Packet = P13 BP=82 Heart=39 05/11/2023 19:34:
34
```

Fig 5.6 Packet Transmission Process

In above screen in blue colour text we can see verification output and similarly you can select patient sensor and send to hospital for storage and below is another example.

5.7 Signature Generation and Encryption


Blockchain for the Management of Internet of Things Devices in the Medical Industry
Generate Medical IOT Sensors
Generate Private & Public Keys
Start Simulation
Verification Time Chart
Read Data from Blockchain
P5
P6
P11
P13
P18
P19
P9
P12
Hospital
P7
P4
P1
P10
P8
P15
P14
P3
P16
P2
P17
Type here to search
27°C
ENG
19:37
05-11-2023

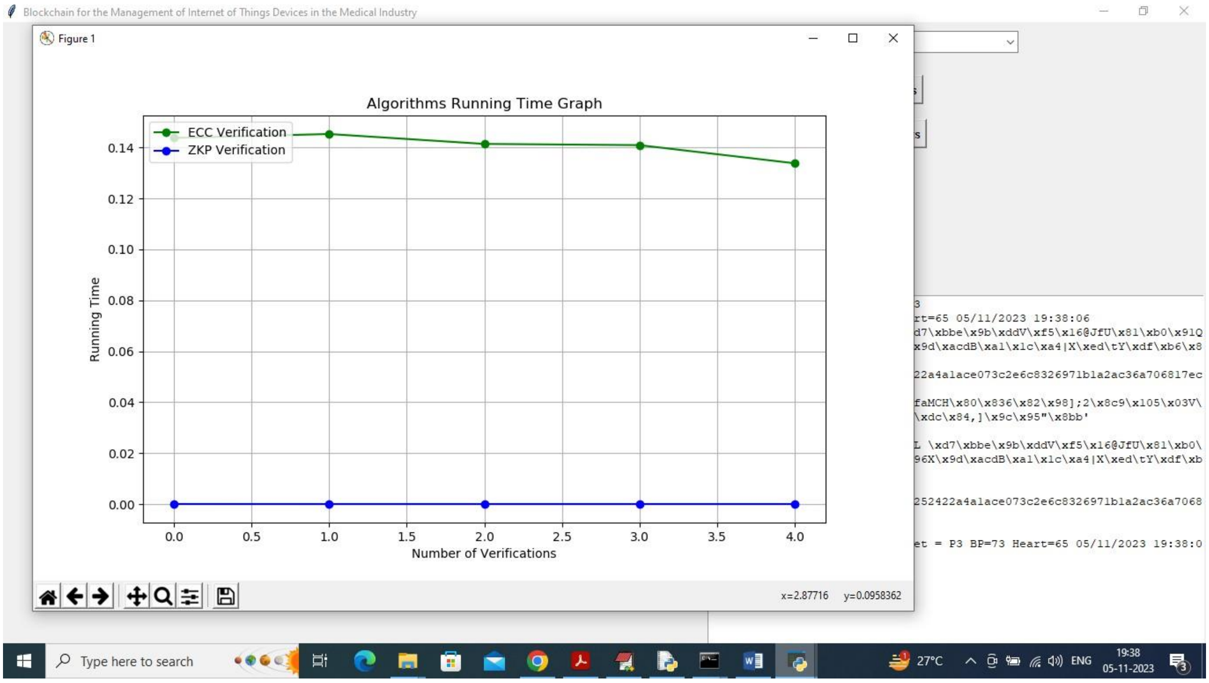
IOT Sensor ID: 8

```
Selected Patient Sensor is : 8
Sending Packet : P8 BP=123 Heart=83 05/11/2023 19:37:40
ECC Signature : b']\xa04\x1fk\x9f4\x83\x16\xf2\xfem\n"\x02\x08\x1e\x7f
1B\xcf<fxc`:\xc7\xa3\xfe\x14C\x90(W\xae\xc0{\xc3Y8\xc4\xe0\x1a\xe0\xd4
2\x9f'
ZKP Signature : 6aa80748a9b5c8163ec9c7b53c35400a6a47469ecb0fc010651f5a
7ce196bf1b
AES Encrypted Data : b'\x16\xf1MCH\x80\x857\x91\xf0p?!\x8ap\x1b8\x10F\
xe2o\x07\x05\xbfK\xbd\xe9\xca\xcf\x95$^\x95\x9e/\x81`p'
```

Fig 5.7 Signature Generation and Encryption

In above screen can see another output and now click on ‘Verification Time Chart’ button to get below graph.

5.8 Verification Time Comparison Chart

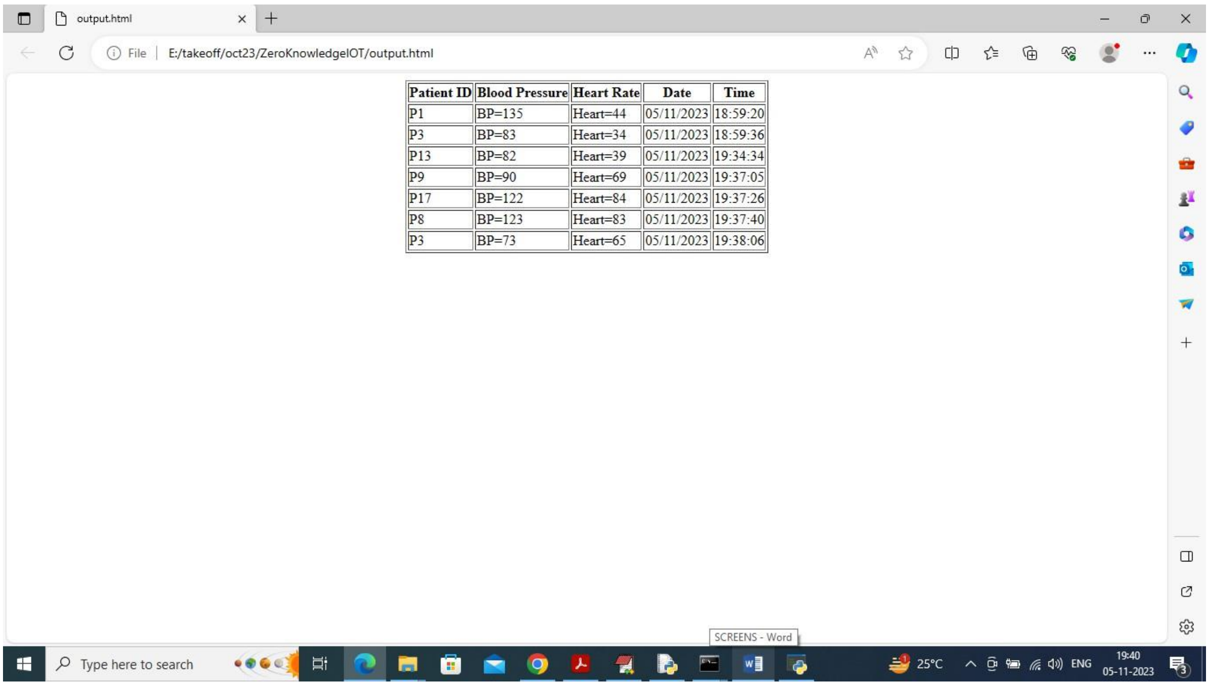


```
3
rt=65 05/11/2023 19:38:06
d7\xbbe\x9b\xddV\xf5\x16@JfU\x81\xb0\x91Q
x9d\xacdB\xa1\x1c\xa4|X\xed\tY\xdf\xb6\x8

22a4a1ace073c2e6c8326971b1a2ac36a706817ec

faMCH\x80\x836\x82\x98];2\x8c9\x105\x03V\
\xdc\x84,]\x9c\x95"\x8bb'

L \xd7\xbbe\x9b\xddV\xf5\x16@JfU\x81\xb0\
96X\x9d\xacdB\xa1\x1c\xa4|X\xed\tY\xdf\xb


252422a4a1ace073c2e6c8326971b1a2ac36a7068


et = P3 BP=73 Heart=65 05/11/2023 19:38:0
```

Fig 5.8 Verification Time Comparison Chart

In above graph x-axis represents ‘Number of Verification’ and y-axis represents computation time. Blue line represents ZKP verification time and green line represents ECC verification time and in both algorithms ZKP (zero knowledge) us taking less time. Now close above graph and then click on ‘Read Data from Blockchain’ button to read all patients data and display in below page.

5.9 Blockchain Data Retrieval



Patient ID	Blood Pressure	Heart Rate	Date	Time
P1	BP=135	Heart=44	05/11/2023	18:59:20
P3	BP=83	Heart=34	05/11/2023	18:59:36
P13	BP=82	Heart=39	05/11/2023	19:34:34
P9	BP=90	Heart=69	05/11/2023	19:37:05
P17	BP=122	Heart=84	05/11/2023	19:37:26
P8	BP=123	Heart=83	05/11/2023	19:37:40
P3	BP=73	Heart=65	05/11/2023	19:38:06

Fig 5.9 Displaying stored patient data from Blockchain

In above screen can see all patients details saved in Blockchain. Similarly by following above screens you can secure and store patient data in Blockchain and then read back.

6. VALIDATION

The validation of this project primarily relies on comprehensive testing and the use of well-defined test cases to ensure the accuracy, reliability, and effectiveness of the blockchain-enabled IoT management system within the medical industry. The validation process includes multiple stages, such as functional validation, security checks, system performance evaluation, and real-world testing in simulated hospital environments. By following a structured validation methodology, the system is tested rigorously to ensure secure data handling, access control, and accurate integration of blockchain and IoT technologies in a healthcare setting.

6.1 INTRODUCTION

The system is validated by simulating real-world scenarios involving various healthcare roles such as doctors, nurses, administrators, and patients interacting with IoT devices (e.g., medical sensors). These sensors transmit health data such as heart rate and oxygen levels to the blockchain system through a secure medical cloud. To ensure robust and generalized performance, modular testing is applied to every major component—data acquisition, encryption, blockchain logging, and smart contract validation.

The blockchain’s inherent immutability and audit trail capabilities are confirmed by performing traceability checks on stored transactions. he blockchain ledger can handle continuous streams without failures. The smart contract logic accurately enforces security policies. The system behaves reliably when scaled to multiple hospitals or sensor nodes. This real-time deployment simulation verifies that the system is stable, secure, and scalable for practical implementation in the medical sector.

6.2 TEST CASES

TABLE 6.2.1 UPLOADING SENSOR DATA

Test case ID	Test case name	Purpose	Test Case	Output
1	Uploads Sensors Dataset.	To upload real-time patient data	The hospital sensor uploads patient vitals (e.g., heart rate, oxygen level)	Dataset successfully uploaded.

TABLE 6.2.2 BlOCKCHAIN RECORDING

Test case ID	Test case name	Purpose	Input	Output
1	Blockchain logging	To verify data is recorded on blockchain	Patient sensor data	Data successfully stored on chain.
2	Immutability test	To check if blockchain prevents tampering	Attempt to alter blockchain record	Alteration denied

TABLE 6.2.3 SMART CONTRACT ACCESS CONTROL

Test case ID	Test case name	Purpose	Input	Output
1	Authorized access	To check access for authorized user	Doctor requests patient data	Access granted
2	Unauthorized access	To check denial for unauthorized user	Random user attempts access	Access denied

7. CONCLUSION & FUTURE ASPECTS

In conclusion, the project has successfully achieved its objectives, showcasing significant progress and outcomes. The implementation and execution phases were meticulously planned and executed, leading to substantial improvements and insights. Looking ahead, the future aspects of the project hold immense potential. Future developments will focus on expanding the scope, integrating new technologies, and enhancing sustainability. These advancements will not only strengthen the existing framework but also open new avenues for growth and innovation, ensuring the project remains relevant and impactful in the long term. This strategic approach will drive continuous improvement and success.

7.1 PROJECT CONCLUSION

In conclusion, the integration of blockchain technology in the management of Internet of Things (IoT) devices within the medical industry presents a transformative opportunity to enhance data security, interoperability, and patient care. By leveraging the decentralized and immutable nature of blockchain, healthcare stakeholders can ensure the integrity and authenticity of sensitive patient data collected from various IoT devices. This secure framework facilitates seamless communication between devices, reduces the risk of data breaches, and fosters trust among patients and healthcare providers.

The transparent nature of blockchain enables efficient tracking and auditing of medical devices, ensuring compliance with regulatory standards. As the medical industry increasingly relies on IoT technology, adopting blockchain solutions not only enhances operational efficiency but also paves the way for innovative healthcare models that prioritize patient safety and data privacy. Ultimately, the synergy between blockchain and IoT has the potential to revolutionize healthcare delivery, creating a more connected and secure environment for both patients and providers.

7.2 FUTURE ASPECTS

The integration of blockchain technology with the Internet of Things (IoT) in the medical industry holds significant promise for enhancing data management, security, and interoperability. As healthcare increasingly relies on connected devices for patient monitoring, diagnostics, and treatment, the need for a robust and transparent system to manage the vast amounts of data generated becomes crucial. Blockchain can provide a decentralized and immutable ledger that ensures the integrity of medical data collected from IoT devices, reducing the risk of tampering or unauthorized access.

Moreover, blockchain's inherent capabilities for smart contracts can automate and streamline processes such as patient consent management, data sharing agreements, and insurance claims processing, thereby improving efficiency and reducing administrative burdens. This technology also enables secure sharing of patient data across different healthcare providers, fostering collaboration and better informed clinical decisions while ensuring compliance with privacy regulations like HIPAA and GDPR. Looking ahead, the convergence of blockchain and IoT in healthcare could lead to innovations such as real time tracking of medical supplies, enhanced remote patient monitoring, and improved supply chain transparency, ultimately driving better patient outcomes. As regulatory frameworks evolve and technological advancements continue, the potential for blockchain to revolutionize the management of IoT devices in the medical field will likely expand, paving the way for more resilient and patient centered healthcare systems.

8. BIBLIOGRAPHY

8.1 REFERENCES

1. Kouicem, D., M. A. K., & H. A. B. (2018). "Blockchain for Internet of Things: A Survey." IEEE Communications Surveys & Tutorials , 21(4), 3116 3145.

2. M. J. H. et al. (2020). "Blockchain Technology in Healthcare: A Comprehensive Review." IEEE Access , 8, 192569 192582.

3. Rehman, M. H. et al. (2020). "Blockchain based Secure Framework for Smart Healthcare System." IEEE Access , 8, 153453 153464.

4. H. J. et al. (2021). "Secure and Efficient Medical Data Sharing System Using Blockchain Technology." Journal of Medical Systems , 45(1), 1 9.

5. G. A. et al. (2019). "The Role of Blockchain in Internet of Medical Things." International Journal of Medical Informatics , 132, 103946.

6. Singh, S. et al. (2020). "Blockchain for Healthcare: A Systematic Review." Journal of Biomedical Informatics , 103, 103380.

7. Y. H. et al. (2019). "A Framework for Secure Medical Data Sharing Using Blockchain Technology." Journal of Medical Internet Research , 21(2), e14094.

8. D. S. et al. (2020). "Blockchain for Health Data Sharing: A Systematic Review." BMC Medical Informatics and Decision Making , 20(1), 1 17.

9. Kumar, V., et al. (2020). "Blockchain in Healthcare: Applications, Opportunities, and Challenges." Journal of King Saud University Computer and Information Sciences , 32(4), 486 497.

10. Bertino, E., & Islam, N. (2020). "IoT Security and Privacy: A Blockchain Perspective." IEEE Security & Privacy , 18(1), 60 63.

8.2 GITHUB LINK

https://github.com/mahesh8075/Blockchain-for-the-Management-of-IOT-devices-in-medical-industry