People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research



University of Echahid Cheikh Larbi Tebessi

Faculty of Exact Sciences And Natural And Life Sciences

Department of Mathematics and Computer Science

MASTER'S THESIS

to obtain the diploma of Master degree in Computer Science

Option: Networks and computer security

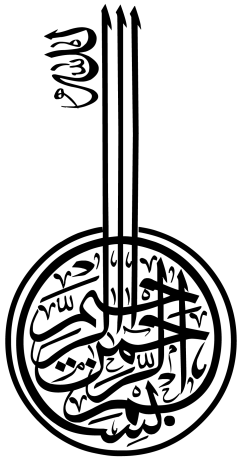
Blockchain Technology for Product Traceability: Application to Pharmaceutical Products.

Realized by:
Khimeche Aimene

Under supervision of:
Mekhaznia Tahar

Jury President:	Bennour Akram
Examiner:	Sahraoui Abdelatif

2025



Acknowledgments

I would like to express my deepest gratitude to my supervisor, Mekhaznia Tahar, for his constant support, guidance, and motivation during the period of this thesis. His expertise and positive criticism were of great help in deciding the ultimate outcome of my research. I owe a great debt to my family, especially my father and mother, whose patience, love, and unconditional support have been my greatest strength in this process. They had faith in me, and that emboldened me to overcome any hurdle. I would also like to thank my friends for their support.

ملخص

تناقش هذه الأطروحة تصميم وتنفيذ نظام يعتمد على تقنية البلوكشاين لتتبع وتعقب المنتجات الصيدلانية. الهدف هو توفير الشفافية والنزاهة والأمان في تتبع المنتج عبر سلسلة التوريد. إن الأنظمة الحالية مركزية وبالتالي فهي عرضة للتلاعب وانعدام الكفاءة. ولمعالجة هذه المشكلات، نقترح تنفيذ تطبيق لا مركزي باستخدام تقنيات الويب الحديثة مثل React لتطوير الواجهة الأمامية، و Express.js للواجهة الخلفية، و PostgreSQL لقاعدة البيانات، وشبكة البلوكشاين باستخدام مكتبات ويب خاصة لتمكين الاتصال والإجماع من نواة إلى نواة دون الحاجة إلى سلطة مركزية.

تتيح هذه العملية للمستخدمين إنشاء سجلات المنتج والمصادقة عليها وحفظها على دفتر مشترك بحيث يتم تسجيل كل مرحلة من مراحل دورة حياة المنتج بشكل لا رجعة فيه. تقوم الأنوية بمصادقة المعاملات وتعدين blocks المنشورة على الشبكة بشكل جماعي.

تعمل هذه العملية على تعزيز سلامة سلاسل توريد الأدوية من خلال القضاء على التزييف والاحتيال والتلاعب غير المصرح به بالمنتجات.

يوضح النموذج الأولي الذي تم إنشاؤه هنا فعالية الجمع بين تقنيات البلوكشاين والويب نحو إمكانية التتبع الآمنة والقابلة للتطوير. يساهم هذا العمل في تطوير حلول إدارة سلسلة التوريد اللامركزية، والأهم من ذلك في الصناعات التي تكون فيها الثقة وإمكانية التتبع ضرورية.

الكلمات المفتاحية: (بلوكشاين، تتبع المنتج، سلسلة التوريد الصيدلية، تطبيق لا مركزي، غير قابل للتغيير)

Abstract

This thesis discusses the design and implementation of a blockchain-based system for pharmaceutical product tracking and tracing. The objective here is to give transparency, integrity, and security in tracking the product across the supply chain. Current systems are centralized and hence prone to tampering and inefficiency. To address these issues, we recommend the implementation of a decentralized application using modern web technologies like React for frontend development, Express.js for the backend, PostgreSQL for database, and a blockchain network was implemented using special web libraries to enable peer-to-peer communication and consensus without the need for a central authority.

The process allows users to create, authenticate, and transfer product records on a shared ledger such that each stage of the life cycle of a product is recorded irretrievably. Peers authenticate transactions and mine blocks published on the network collectively. The process enhances the integrity of pharma supply chains by eliminating counterfeiting, fraud, and unauthorized product tampering.

The prototype created herein demonstrates the effectiveness of the combination of blockchain and web technologies towards secure and scalable traceability. This work contributes to the advancement of decentralized supply chain management solutions, most importantly in industries where trust and traceability are a necessity.

Keywords: (Blockchain, Product Traceability, Pharmaceutical Supply Chain, Decentralized Application, Tamper-Proof)

Résumé

Cette thèse traite de la conception et de la mise en œuvre d'un système basé sur la blockchain pour le suivi et la traçabilité des produits pharmaceutiques. L'objectif est d'assurer la transparence, l'intégrité et la sécurité du suivi du produit dans toute la chaîne d'approvisionnement. Les systèmes actuels sont centralisés et donc sujets à la manipulation et à l'inefficacité. Pour résoudre ces problèmes, nous recommandons l'implémentation d'une application décentralisée utilisant des technologies web modernes comme React pour le développement du frontend, Express.js pour le backend, PostgreSQL pour la base de données, et un réseau blockchain a été mis en œuvre à l'aide de bibliothèques Web spéciales pour permettre la communication et le consensus entre pairs sans avoir besoin d'une autorité centrale.

Le processus permet aux utilisateurs de créer, d'authentifier et de transférer des dossiers de produits sur un registre partagé de sorte que chaque étape du cycle de vie d'un produit soit enregistrée irrémédiablement. Les pairs authentifient collectivement les transactions et les blocs de mine publiés sur le réseau. Le processus améliore l'intégrité des chaînes d'approvisionnement pharmaceutiques en éliminant la contrefaçon, la fraude et les manipulations non autorisées de produits.

Le prototype créé ici démontre l'efficacité de la combinaison des technologies blockchain et web pour une traçabilité sécurisée et évolutive. Ce travail contribue à l'avancement des solutions décentralisées de gestion de la chaîne d'approvisionnement, plus particulièrement dans les industries où la confiance et la traçabilité sont une nécessité.

Mots clés : (Blockchain, traçabilité des produits, chaîne d'approvisionnement pharmaceutique, Application décentralisée, inviolable)

Table of Contents

General Introduction **1**

1 Introduction to Blockchain Technology. **3**

1.1 Introduction: . . . 3
1.2 Blockchain Technology: . . . 3
1.3 Blockchain Features: . . . 4
1.3.1 Decentralization: . . . 4
1.3.2 Immutability: . . . 4
1.3.3 Security: . . . 4
1.3.4 Transparency: . . . 4
1.3.5 Traceability: . . . 4
1.4 Blockchain Components: . . . 5
1.4.1 Nodes: . . . 5
1.4.2 Transaction: . . . 6
1.4.3 Block: . . . 7
1.4.4 Hash: . . . 7
1.4.5 Merkle Tree: . . . 8
1.4.6 Consensus Mechanisms: . . . 9
1.4.7 Block Creation: . . . 9
1.5 Blockchain Types: . . . 12
1.5.1 Public Blockchain: . . . 12
1.5.2 Private Blockchain: . . . 12
1.5.3 Hybrid Blockchain: . . . 12
1.5.4 Consortium Blockchain: . . . 13
1.6 Conclusion: . . . 13

2 State of the Art **14**
2.1 Introduction: . . . 14
2.2 Pharmaceutical Supply Chain Fundamentals: . . . 14
2.2.1 What is Pharmaceutical Supply? . . . 14
2.2.2 How Pharmaceutical Supply Chain Works: . . . 15
2.2.3 Economic Importance of Pharmaceutical Supply: . . . 15
2.3 Blockchain on Pharmaceutical Supply Chains: . . . 15
2.4 Literature Review: . . . 16
2.5 Synthesis and Discussion: . . . 17
2.6 Conclusion: . . . 18

3 Contributions **20**
3.1 Introduction: . . . 20
3.2 Theoretical Proposal: . . . 20
3.2.1 Project Description and Objectives: . . . 20
3.2.2 System Architecture: . . . 22
3.3 Implementation: . . . 26
3.3.1 Technologies used for implementation: . . . 26
3.3.2 Database Implementation: . . . 28
3.3.3 Blockchain Implementation: . . . 30
3.3.4 Tamper-Proofing and Chain Integrity: . . . 37
3.4 Experimental Process: . . . 38
3.4.1 Test Environment: . . . 38
3.4.2 Data Used: . . . 38
3.4.3 Obtained Results: . . . 38
3.4.4 Functional Overview: . . . 38
3.5 Conclusion: . . . 46

General Conclusion **47**

Bibliography **48**

List of Figures

1.1 Blockchain Example 3
1.2 Blockchain Network 6
1.3 A mempool hold valid transactions 7
1.4 Merkle Tree Hashing 8
1.5 Example of Three Transactions 10
1.6 Merkle Tree of the Three Transactions 10
1.7 Header of the Block 11

3.1 Global use case 22
3.2 System Architecture 23
3.3 Network Architecture 23
3.4 Process Flow Design 26
3.5 Entity Relationship Diagram 29
3.6 Sign in Page 39
3.7 Dashboard Page 40
3.8 Create Product 1 41
3.9 Create Product 2 41
3.10 Transfer Product 42
3.11 Transfer Product 42
3.12 Admin Dashboard 43
3.13 Search Page 44
3.14 Product Statistics Page 45
3.15 Blockchain Visualization Page 45

List of Tables

2.1 Comparison of Literature on Blockchain-based Pharmaceutical Supply Chain Solutions 17

3.1 Users Table in the Database 28
3.2 Products Table in the Database 29
3.3 Overview of technologies used in the system implementation 36

General Introduction

In recent years, the pharmaceuticals product sector faces many challenges, the most important of which is counterfeiting (over 1.5 million counterfeit medicines were seized at Paris's Charles de Gaulle Airport, destined for Algeria [1]). Such incidents highlight the urgent need for enhanced traceability and trust in the supply chain. Blockchain technology presents a promising solution by providing a tamper-proof and transparent record of product movements. This project leverages blockchain to create a secure, transparent system for tracking pharmaceutical products, addressing real-world needs for trust and transparency.

The pharmaceutical sector's exposure to recurring problems such as counterfeiting and lack of oversight (Counterfeit versions of at least 104 Pfizer medicines have been confirmed in more than 116 countries.[2] and in terms of geographical distribution, North America recorded 2,442 counterfeit pharmaceutical seizures, followed by Asia Pacific with 1,747 cases. [3]). All of this has led to the development of fairly effective solutions, although they have not solved all the important problems. Those solutions often work with centralized databases, which have been implemented to improve traceability and security, but these solutions are vulnerable to problems due to their centralized nature,such as data tampering and single points of failure, because traditional systems often store data in centralized databases, making them vulnerable to breaches and inconsistencies.[4] This reinforces the need for a product tracking system without fear of problems such as tampering.

In this work, we propose a blockchain-based traceability system that is tailored for the pharmaceutical supply chain. The proposed system is a web-based application that offers a user-friendly interface, enabling stakeholders to register pharmaceutical products, and track them.

Because of the decentralized nature of blockchain technology, the system is secure and tamper-proof. This architecture prevents unauthorized modifications and reduces the risk

of counterfeit drugs entering the supply chain. By leveraging key features of blockchain — such as distributed ledgers, cryptographic hashing, and consensus mechanisms — the system guarantees data integrity, traceability, and trust among all participants.

This thesis is organized as follows: In the first chapter, we will discuss general information about blockchain technology. This includes definitions, essential features such as decentralization, immutability, and security, as well as key components like nodes, transactions, blocks, and consensus mechanisms. We also explain how blocks are created and explore the various types of blockchain systems, including public, private, hybrid, and consortium models.

In the second chapter, we present the state of the art by reviewing existing literature and related works. This chapter provides a synthesis and discussion of current blockchain applications and research trends, highlighting their strengths and limitations, and identifying gaps that motivate our contribution.

The third chapter is dedicated to our contributions. We begin by outlining the theoretical proposal, including a description of the project's objectives and the proposed system architecture. Then, we move to the implementation phase, detailing the technologies used, the design of the database, and the blockchain implementation. Special attention is given to ensuring tamper-proofing and maintaining chain integrity. We conclude with a functional overview of the system to illustrate its practical application.

Chapter 1

Introduction to Blockchain Technology.

1.1 Introduction:

Blockchain is a secure and decentralized technology that records and manages data transparently. This chapter explores its definition, key features like decentralization and security, and essential components such as nodes, ledgers, and consensus mechanisms. It also classifies blockchain types—public, private, hybrid, and consortium—highlighting their strengths and weaknesses. Finally, the chapter explains how blockchain works, ensuring security and trust across a distributed network.

1.2 Blockchain Technology:

Blockchain is a distributed database where data is stored in blocks tied together in a chain. Data in any particular block cannot be altered or deleted without the network's consensus. This makes blockchain a consistent ledger to track products, payments, accounts, and transactions. See figure below (1.1).

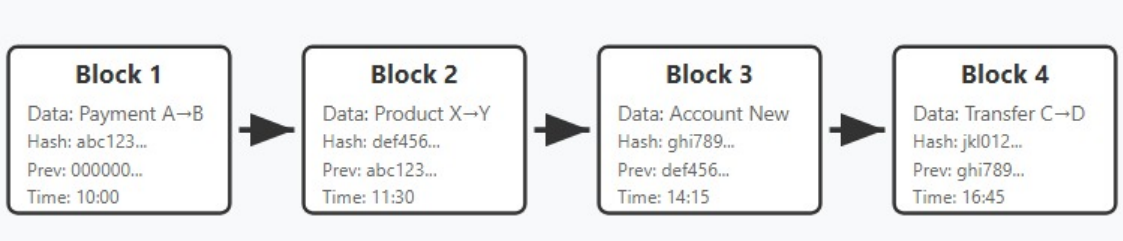


Figure 1.1: Blockchain Example

1.3 Blockchain Features:

1.3.1 Decentralization:

Blockchain operates on a peer-to-peer network where no single entity controls it. This reduces the risk of central failures and enhances transparency. The blockchain consists of peer-to-peer blocks, each of which has the ability to record and store all transactions. Technically speaking, information is automatically shared and distributed between nodes without any third-party intervention. In this decentralized system, all participants and nodes are active to join the activities and transactions. Unlike the centralized authority of conventional mechanisms, decentralization is a distinct feature of the blockchain and attracts more attention.[5]

1.3.2 Immutability:

It is impossible to tamper with any transaction on the blockchain. If any transaction is wrongly executed, a new transaction must be issued to right the wrongs of the previous transaction.[6] So data in blockchain cannot be edited or deleted once recorded, increasing trust and preventing fraud.

1.3.3 Security:

Blockchain ensures security through a combination of cryptographic techniques and consensus mechanisms. Each block contains a cryptographic hash of the previous one, making the data tamper-evident. Transactions are digitally signed, ensuring authenticity and preventing fraud. Because the ledger is distributed across many nodes, it's resistant to attacks and system failures.

1.3.4 Transparency:

Blockchain allows participants to check records, so all participants share records and query data in a decentralized structure and each participant can query the records in the blockchain to make the information in the distributed system transparent and consistent. Each transaction data of a distributed system is open and reliable.[5]

1.3.5 Traceability:

Every block is tied to previous ones, forming a chain of events, which helps track product history, ownership, or data flow.

The blockchain uses timestamps to identify and to record each transaction. This allows the node to keep the order of transactions and to make the data traceable.[5]

1.4 Blockchain Components:

1.4.1 Nodes:

Nodes are computers running specialized software that allows them to interact with the blockchain. Nodes connect to each other (peer-to-peer connection) to forming a blockchain network as shown in Figure (1.2); there are two main types of nodes:

1.4.1.1 Full Nodes:

Keep a complete record of all transactions and blocks, and validate every block and transaction against consensus rules.

How They're Built:

- They're computer that require large disk space (SSD preferred), and enough RAM and CPU for validation and stable internet connection,. For example to run a bitcoin full node you should have at least 7 gigabytes of free disk space, accessible at a minimum read/write speed of 100 MB/s and 2 gigabytes of memory (RAM) and a broadband Internet connection with upload speeds of at least 400 kilobits (50 kilobytes) per second. [7]
- They must download the entire blockchain.

Advantages:

- Full nodes independently verify all data, making them the most secure type of node.

Weaknesses:

- Special requirements (storage, bandwidth, and computation).
- New full nodes take time to download and verify the blockchain.

1.4.1.2 Partial Nodes (Lightweight Nodes):

Only sotore block headers and rely on full nodes to fetch full data as needed.

How They're Built:

- Use a lightweight client (e.g., Electrum for Bitcoin), can used also in phones.
- Only store headers of blocks (80 bytes per block).

Advantages:

- Requires less hardware power.
- New partial nodes sync faster then full nodes since only block headers are downloaded.

Weaknesses:

- Depend on full nodes for verification, what make it less secure and less trusted.

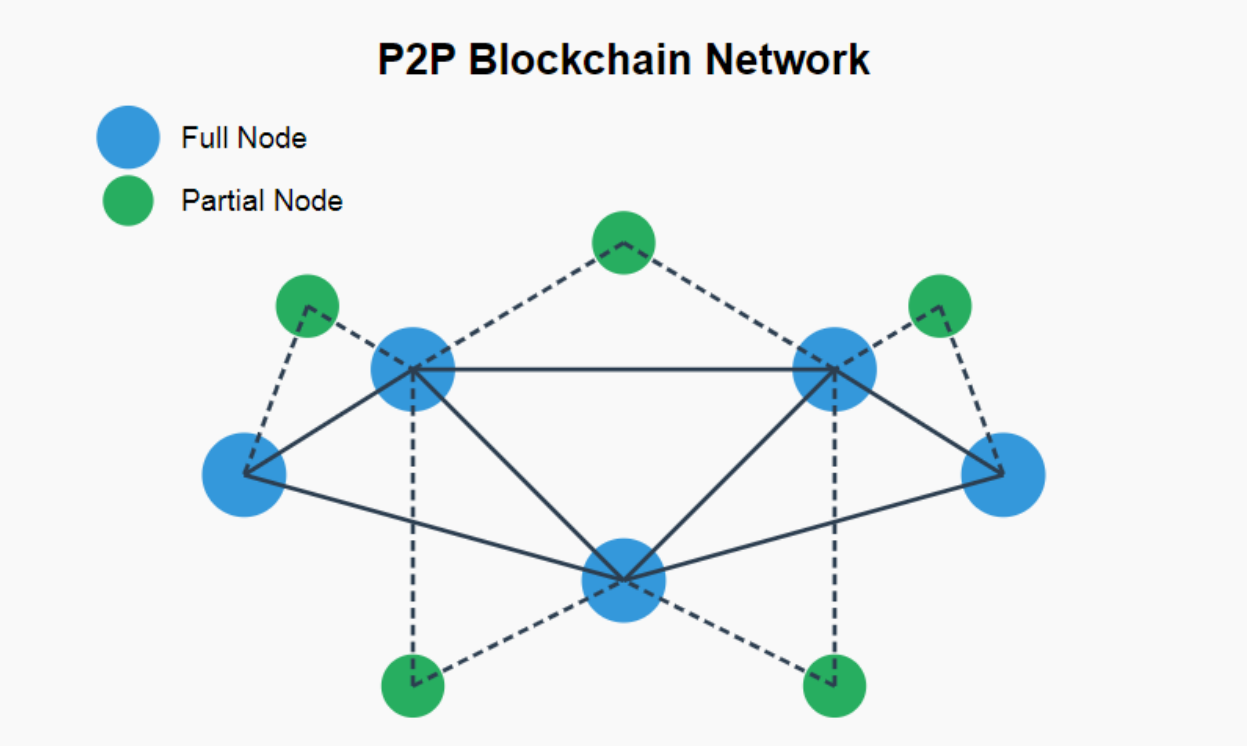


Figure 1.2: Blockchain Network

1.4.2 Transaction:

A transaction is a record of data transfer between two or more participants on a blockchain network . When a user broadcasts a transaction to the network, full nodes receive it and validate it (by ensuring that data is correctly formatted), and then add the transaction to the mempool (a temporary storage area that holds transactions not yet

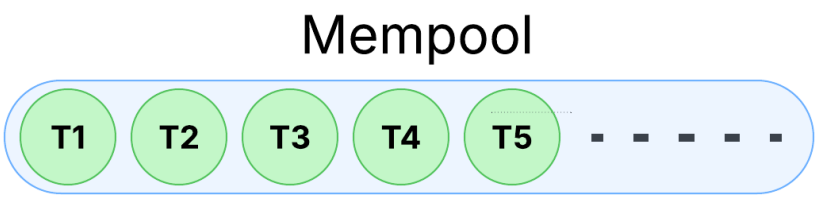


Figure 1.3: A mempool hold valid transactions

included in a block). For example if someone wants to send 0.5 BTC to another person, the wallet broadcasts the transaction to the Bitcoin network. Once it received and validated, its added to the mempol as shown in Figure (1.3), waiting to included in a block and get mined by miners.

1.4.3 Block:

A block is a container of data that is appended to the blockchain (a distributed ledger) in a linear, chronological order. Each block is linked to the previous one using cryptographic hashes.

1.4.3.1 Structure of a Block:

- **Block Header:**The block header holds metadata about the block. The metadata can be different for each blockchain but most of them have similar structure (the blockchain version, the hash of data (Merkle Root), the previous block hash, the timestamp, and the nonce).
- **Data:** A block contains one or list of valid transactions or records.
- **Block Hash:** Cryptographic hash (e.g., SHA-256) of the block header. It uniquely identifies the block and linking it to the previous block.

1.4.4 Hash:

The hash is the backbone of blockchain technology. Because hashing ensures:

- **Data Integrity:** If any part of a transaction or block changes, even a single character, the hash completely changes.
- **Block Linking:** Each block contains the hash of the previous block. If someone changes a block, its hash changes, and the link breaks.
- **Efficient Verification:**Hashes allow fast verification of data.

The hash function (SHA-256) for example is secure because it's deterministic (same input gives same output), is one-way (can't reverse the hash), is collision-resistant (very unlikely to produce the same hash for different inputs), and small changes alter the hash.

1.4.4.1 How SHA-256 Works:

- 1- Converting the input data to binary.
- 2- Padded the binary number with zeros so its length will be equal to 448 modulo 512.
- 3- Complete the last 64 bits with the length of the original data.
- 4- Divides the new binary number into 512-bit blocks.
- 5- Each block goes through complicated 64 rounds of processing using nonlinear functions and bitwise operations like rotate, shift, XOR, AND, OR, etc, in addition to repeatedly dividing, mixing and collecting blocks.
- After all of this a 256-bit (64-character hex) value is returned.

Example: Hello=185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969

1.4.5 Merkle Tree:

Also known as a hash tree, it is a binary tree structure used in blockchain systems to securely summarize and verify large sets of transactions. It works by taking all the transactions in a block, hashing them by a hash function (e.g., SHA-256), and then repeatedly hashing pairs of hashes until only one hash remains (called Merkle Root) as shown in Figure (1.4).

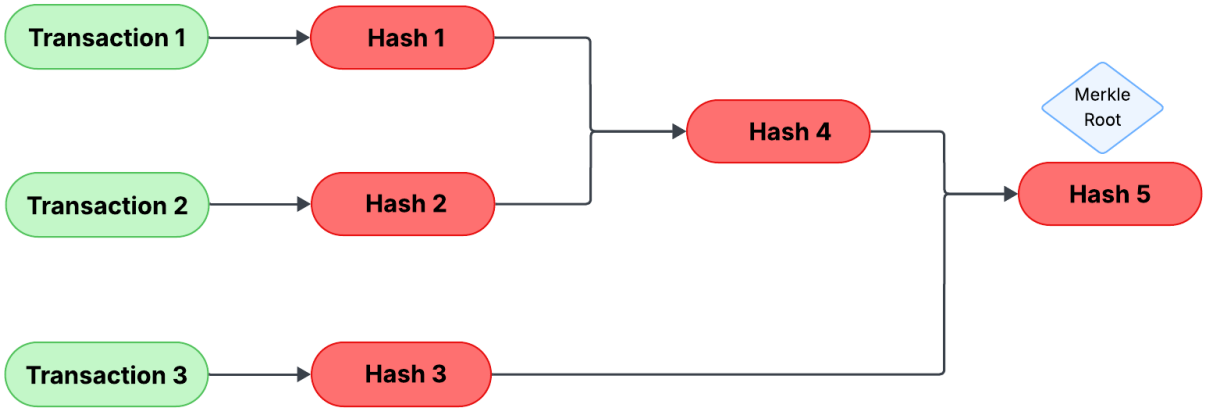


Figure 1.4: Merkle Tree Hashing

1.4.6 Consensus Mechanisms:

Before adding the new block to the blockchain, the network must reach consensus that the block is valid, and this is done through a consensus mechanism. Consensus mechanisms are protocols that allow nodes to agree on valid blocks without a central authority. They ensure security, consistency, and trust in the blockchain. Here are some consensus algorithms:

1.4.6.1 Proof of Work (PoW):

The miners try to solve intricate cryptographic problems (often with hashes). The miner who discovers a valid solution produces a new block and announces it to the network. The other nodes confirm the block's validity to including it in their own versions of the blockchain. This process requires significant computing power and energy.

1.4.6.2 Proof of Stake (PoS):

Instead of mining, validators are selected to propose and confirm blocks based on how much cryptocurrency they've locked (or staked) in the network. The more they stake, the higher their chances of being chosen. Misconduct can result in slashing, where a portion of the staked funds is lost.

1.4.6.3 Proof of Capacity:

Uses available hard drive space instead of computation power, allowing validators with more storage to gain a higher probability of block validation.

1.4.6.4 Proof of Authority:

A few pre-approved, identifiable validators are granted the authority to process transactions and produce blocks. They usually operate under legal or organizational obligations.

There is other consensus algorithms, like Proof of Activity, Proof of Weight, and Leased Proof of Stake. Choosing the right consensus mechanism depends on the network's goals, scalability, and security requirements.

1.4.7 Block Creation:

Example for (Cryptocurrency transaction, PoW Algorithm):

Let's suppose that there are three valid transactions in a mempool as shown in Figure (1.6).



Figure 1.5: Example of Three Transactions

Step One:

Miners pick transactions from mempool and build a Merkle Tree to prepare a block as shown in Figure (1.6).

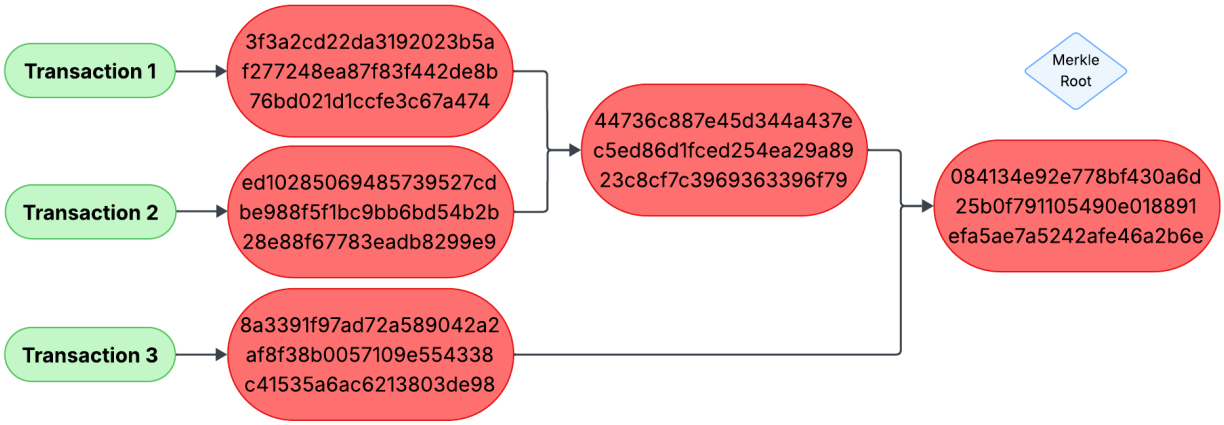


Figure 1.6: Merkle Tree of the Three Transactions

Step Two:

Block header finalization by adding all the metadata of the block as shown in Figure (1.7).

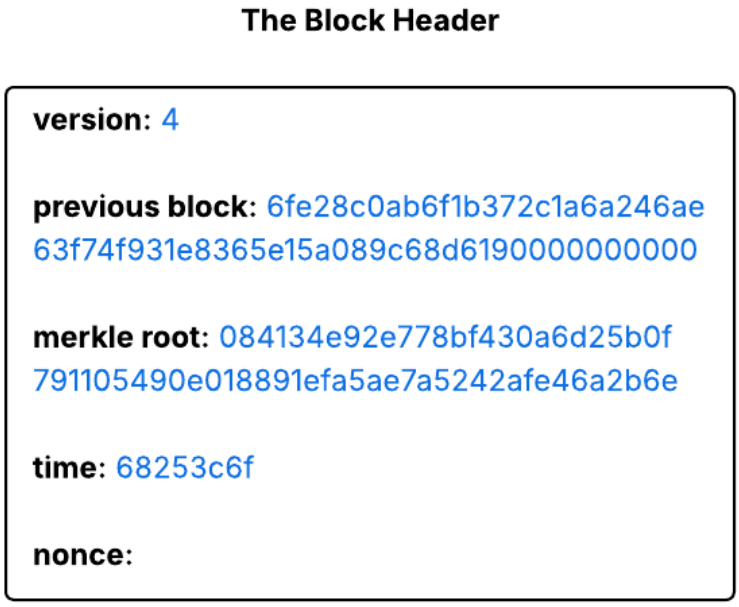


Figure 1.7: Header of the Block

Note: nonce will be calculated in the third step.

Step Three:

Miner starts mining the block with the PoW algorithm, by hashing the header of the block repeatedly and increasing the nonce to find a hash below the target.
The target is a 256-bit number that defines how difficult it is to find a valid hash. The lower the target, the harder it is to find a valid hash.
Bitcoin for example make the target more harder every 2016 block.

Step Four:

When a miner finds a valid hash, he broadcasts the block to the network.
Other nodes verify the block and its PoW.

Step Four:

When a miner finds a valid hash, he broadcasts the block to the network.
Other nodes verify the block and its PoW.
If valid, nodes add the block to their copy of the blockchain.
The miner gets the block reward (new bitcoins + transaction fees).

1.5 Blockchain Types:

1.5.1 Public Blockchain:

A decentralized network open to anyone, operating on a permissionless system.

1.5.1.1 Advantages:

- Trustworthy and secure: Fraud detection ensures safety, and records are widely distributed.
- Anonymity: Users are not required to disclose their identities.
- Decentralization: No single entity controls the network.

1.5.1.2 Disadvantages:

- Energy-intensive: Proof-of-work consumes a lot of power.
- Regulation issue: Hard for governments to control and implement.

1.5.2 Private Blockchain:

A restricted network managed by a single organization or a group of entities.

1.5.2.1 Advantages:

- Fast and scalable: Fewer nodes mean quicker data exchange and flexible network size.
- Enhanced privacy: Suitable for businesses requiring confidentiality.
- Better performance: Since only selected users participate, the network operates efficiently.

1.5.2.2 Disadvantages:

- Security risk: Fewer nodes make manipulation easier.
- Centralized control: A single entity can misuse power.

1.5.3 Hybrid Blockchain:

A combination of public and private blockchains.

1.5.3.1 Advantages:

- Stronger security: Hard to hack.

- Customizable and controlled: Organizations decide who participate and what is public.

1.5.3.2 Disadvantages:

- Complex setup: Difficult to implement and maintain.
- Limited transparency: Organization may hide information.

1.5.4 Consortium Blockchain:

A semi-decentralized network managed by a group of organizations.

1.5.4.1 Advantages:

- Fast and secure: Limited participants, speed up transactions while ensuring security.
- Controlled privacy: Only members can access data.

1.5.4.2 Disadvantages:

- Complex governance: Requires coordination among multiple entities.

1.6 Conclusion:

Blockchain is a transformative technology that ensures security, transparency, and decentralization in digital transactions and supply chain. This chapter covered its definition, key features, essential components, and different types, along with an overview of how it operates.

Chapter 2

State of the Art

2.1 Introduction:

This chapter explores the use of blockchain technology in pharmaceutical supply chains traceability systems. We begin by examining the fundamentals of pharmaceutical supply chains, including their definition, how it works, and economic significance. The chapter goes on to explore how blockchain technology can prevent the frailties and problems that are implicated in pharmaceutical supply chains, in particular problems with counterfeiting and transparency.

The core of this chapter is a comprehensive review of the existing literature in blockchain-based pharmaceutical traceability systems. We compare various solutions proposed by researchers, examining their technical realization. Following this literature review, we provide a synthesis and discussion that highlights key insights, remaining gaps, and limitations of current solutions, including cost estimation challenges, scalability, and issues with practical implementation.

2.2 Pharmaceutical Supply Chain Fundamentals:

2.2.1 What is Pharmaceutical Supply?

The pharmaceutical supply chain refers to the series of interconnected activities involved in the sourcing, manufacturing, distribution, and delivery of pharmaceutical products to end-users, including patients, healthcare providers, pharmacies, and hospitals. It plays a critical role in ensuring that medications are available when needed and in the right quantities, ultimately impacting patient health outcomes.[8]

2.2.2 How Pharmaceutical Supply Chain Works:

As noted in a 2005 Kaiser Family Foundation report, the following five pharmaceutical supply chain strategies ensure that drug inventory is readily available for distribution to providers and patients:

- 1- Drugs are manufactured at production sites.
- 2- They are then transferred to wholesale distributors.
- 3- The pharmaceuticals are stocked at various types of pharmacies, including retail and mail-order.
- 4- Pharmacy benefit management companies negotiate prices and process drugs through quality and utilization management checks.
- 5- Finally, pharmacies dispense the drugs to patients, who take them as prescribed.

Researchers note there are many variations on this basic structure of the pharmaceutical supply chain, mainly due to the constantly evolving players. The vital players of the pharmaceutical supply chain network allow it to run smoothly and efficiently. Those players include manufacturers, wholesale distributors, pharmacies, and PBMs.[9]

2.2.3 Economic Importance of Pharmaceutical Supply:

- The global pharmaceutical market size was estimated at US$ 1,573.20 billion in 2023 and is projected to grow to US$ 3,033.21 billion by 2034, rising at a compound annual growth rate (CAGR) of 6.15% from 2024 to 2034. [10]
- , the global pharmaceutical industry supported an additional 45.1 million indirect jobs in other industries along its supply chains.[11]

2.3 Blockchain on Pharmaceutical Supply Chains:

The complexity of pharmaceutical supply chain make it vulnerable to counterfeiting, fraud and other problems. Which make developers and researchers motivates to develop solutions that reduce those risks, starting from traditional solutions to integrate innovative technologies that can enhance transparency, traceability, and security such as Artificial Intelligence and Blockchain. In the recent years solutions that integrate blockchain technology start showing a lot of success, what tell that is a promising solution. In the next section we will discuss about some of those solutions.

2.4 Literature Review:

In this section, we examine significant contributions in the literature of the integration of blockchain technology into the pharmaceutical supply chain. Where has gained growing attention from researchers seeking to address persistent challenges such as counterfeiting, lack of transparency, and poor data traceability. [12] Highlight the potential of blockchain as a decentralized infrastructure to enhance the integrity of supply chains. They introduce an ethereum blockchain-based approach leveraging smart contracts with a multi-layered data storage and validation mechanism that mitigates counterfeiting attacks. [13] Propose a cloud-based and Hyperledger fabric solution to provide a safe and secure environment for information. They expand on the operational dimensions of blockchain in the pharmaceutical sector. They emphasize the importance of serialization and digital authentication in combating counterfeit drugs. By advocating for optimized and automated processes, their work reinforces the value of blockchain in improving both accountability and supply chain efficiency. [14] Underscores the vulnerabilities within complex pharmaceutical supply chains, where multiple actors can introduce risks. They positions blockchain as a reliable tool to improve transparency and safeguard the flow of life-saving medications, particularly in regions prone to falsified drugs. So they propose an ethereum blockchain-based solution that uses smart contracts and data immutability. [15] Propose a blockchain-based traceability framework that incorporates advanced cryptographic techniques to ensure confidentiality and data auditability. Their model eliminates the need for third-party intermediaries, thus enhancing stakeholder trust. [16] Propose multi-layerd blockchain-based leveraging Hyperledger Fabric solution, aiming to enhance security and transparency in product delivery. Their conceptual model highlights the limitations of centralized databases and demonstrates the advantages of decentralized, sensor-driven networks. [17] Focus on a blockchain-secured pharmaceutical distribution model that emphasizes economic efficiency and trust by considering nodes reputation and needs. They align their work with the FDA's blockchain initiatives, demonstrating how distributed ledgers can enhance drug testing documentation and integrity in product distribution.

Comparison of Blockchain-Based Solutions in Pharmaceutical Supply Chains

Reference	Blockchain Platform / Type	Key Features	Strengths	Limitations
(2021)[12]	Ethereum / Public	Smart contracts, multi-layered validation	Strong anti-counterfeiting measures; decentralization	High cost due to public blockchain; requires technical expertise
(2022)[13]	Hyperledger Fabric / Private	Serialization, cloud integration, secure sharing	Efficient authentication; private and permissioned network	Limited scalability; stakeholders not part of core blockchain nodes
(2022)[14]	Ethereum / Public	Improved drug flow security; smart contracts	Increased transparency and traceability	Costly deployment; limited stakeholder involvement
(2021)[15]	Custom / Hybrid	Advanced cryptography, auditability	High data integrity; removes third-party trust	Lack of real deployment; unclear usability
(2022)[16]	Hyperledger / Multi-cloud	Sensor-driven tracking; decentralized control	Real-time monitoring; robust data flow	High system complexity; integration difficulty
(2022)[17]	Private Blockchain / Custom	Node reputation, economic efficiency	Tailored to pharma context; aligns with FDA goals	No interface; lacks testing or deployment results

Table 2.1: Comparison of Literature on Blockchain-based Pharmaceutical Supply Chain Solutions

Summary of Key Insights

The reviewed literature confirms a strong and growing consensus on the potential of blockchain technology to address long-standing problems in pharmaceutical product traceability. Key themes include the importance of decentralization, cryptographic verification, and artificial intelligence for predictive analytics and anomaly detection. However, gaps remain in standardization, scalability, user accessibility, and the seamless interaction between multiple stakeholders. These gaps provide a basis for the development of our system, which aims to combine technical robustness with a user-friendly and role-specific interface tailored to real-world pharmaceutical traceability needs.

2.5 Synthesis and Discussion:

The review of existing literature reveals that blockchain technology holds considerable promise for enhancing traceability in pharmaceutical supply chains. Several works have

demonstrated its potential in improving data integrity, transparency, and security. Key technical contributions include decentralized record-keeping, advanced cryptographic protections.

Despite these advancements, current solutions also expose several limitations. The main two gaps are cost and stakeholders effectiveness in the blockchain network, this happens because of using existing blockchains like ethereum,[12][14][15], most of existing blockchain network are paid, meaning users must pay transaction fees to interact with it, these fees are required for a variety of actions, such as deploying smart contracts, executing contract functions, and transferring tokens, and the cost of these operations varies depending on the complexity of the tasks; additionally, during periods of high network activity, the fees can increase significantly. The second thing is the effectiveness of stakeholders in the blockchain network. In existing blockchain networks (even free one) stakeholders do not forming network. The network formed with other independent nodes; as a result, typical stakeholders are not directly involved in the validation or storage process.[13] Many proposed systems remain conceptual or are tested only in controlled environments as [15][17], lacking semi-deployment or real-testing. Usability, accessibility, are overlooked in most of works. Absence of user interface [17]. Experience needed from users[12]Most solution did not pay attention to scalability and integration complexity specially [16] that provide a multi-cloud complex blockchain architecture.

Our project aims to address these gaps by designing a blockchain-based traceability system that emphasizes not only security and transparency but also practical deployment and actor-specific usability. The system introduces user-friendly role-based interface that requires no experience, tailored to manufacturers, distributors, and pharmacists. By combining technical robustness with an intuitive interface and adherence to traceability principles, our approach builds on the strengths of the literature while directly addressing its limitations.

2.6 Conclusion:

This chapter enters into the use of blockchain technology in pharmaceutical supply chains traceability systems. We begin with an overview of the fundamentals of pharmaceutical supply chains, their definition, operation, and economic significance. The chapter proceeds to discuss how blockchain technology can eliminate the vulnerabilities and problems involved in pharmaceutical supply chains, namely those of counterfeiting and transparency. The core of this chapter is a comprehensive literature review of blockchain-based traceability systems in the pharmaceutical industry. We contrast various methods introduced by researchers, pointing out their approach, and technical implementation.

Following this literature review, we then have a synthesis and discussion that highlights key findings, gaps if any, and limitations of existing solutions.

Chapter 3

Contributions

3.1 Introduction:

This chapter presents the main contributions of our work on designing and implementing a blockchain-based traceability system tailored to the pharmaceutical sector. The aim of this project is to enhance the security, transparency, and efficiency of drug traceability by leveraging the core properties of blockchain technology.

We begin by describing the theoretical foundations and architecture of the system, including the key actors, processes, and technological components. Then, we detail the implementation phase, where the proposed system is developed using modern web technologies and blockchain frameworks. The chapter also includes experimental results and analyses that demonstrate the system's ability to meet the requirements of a secure and verifiable pharmaceutical supply chain.

Our contributions lie not only in proposing a technical solution but also in addressing real-world constraints such as role-based access, decentralized validation, and real-time product tracking through intuitive user interfaces.

3.2 Theoretical Proposal:

3.2.1 Project Description and Objectives:

Our system PharmaBlock is designed to improve the safety and transparency of pharmaceutical products. These are the main objectives that we we kept in mind before building our system:

- Design a user-friendly interface that allow users to interact with the system easily without requiring any experience.
- Develop a web-based application instead of a desktop application to ensure cross-

platform accessibility and ensure that the application can used on any device via browsers.

- Leverage blockchain technology to ensure the immutability, transparency, and traceability of all product transactions across the supply chain.
- Combine blockchain with a relational database (PostgreSQL) to ensure that both users and product saved on secure environment.
- Enable real-time product tracking, where the users can view the current status and full history of any pharmaceutical product.
- Provide role-based access and authentication, to ensure that only authorized users can use actions based on their roles.

The system is only used by users who are involved in the pharmaceutical supply chain. So we have chosen to define three main actors in the system, as shown in Figure (3.1):

- **Admin:** Validate product creation and transfer, and view blockchain and products history and statistics.
- **Manufacturer:** Registers new pharmaceutical products and track it.
- **Regular User (e.g. distributor, supplier, etc.).:** Receives products from the manufacturer or another entity and forwards them to pharmacies or another entity, he can also track products.

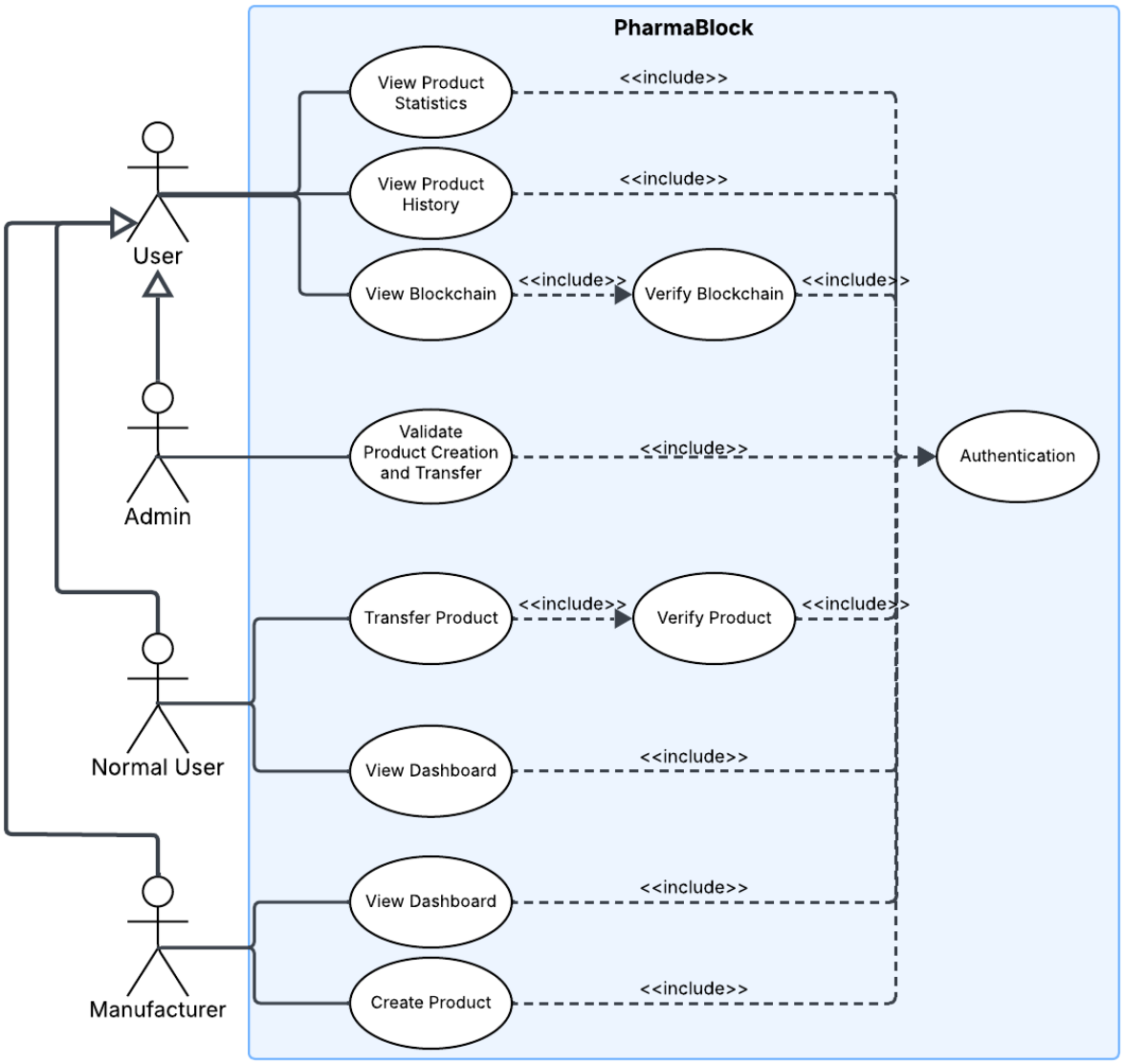


Figure 3.1: Global use case

3.2.2 System Architecture:

The proposed system is based on a multi-layer architecture composed of three main components: the frontend, backend with postgesql database, and blockchain. Each layer is responsible for specific tasks that contribute to the overall functionality of the blockchain-based pharmaceutical product traceability system as shown in Figures (3.2) (3.3).

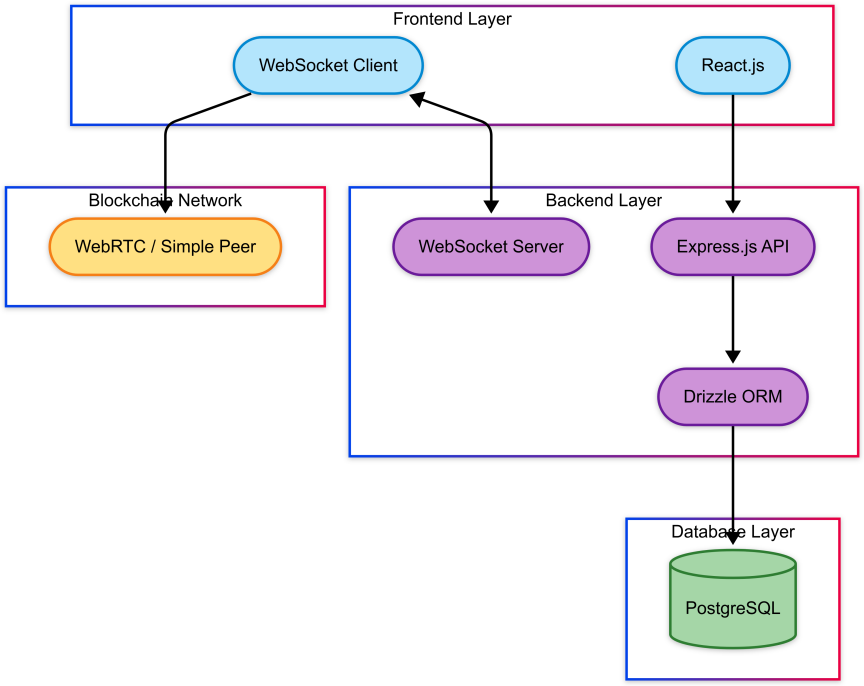


Figure 3.2: System Architecture

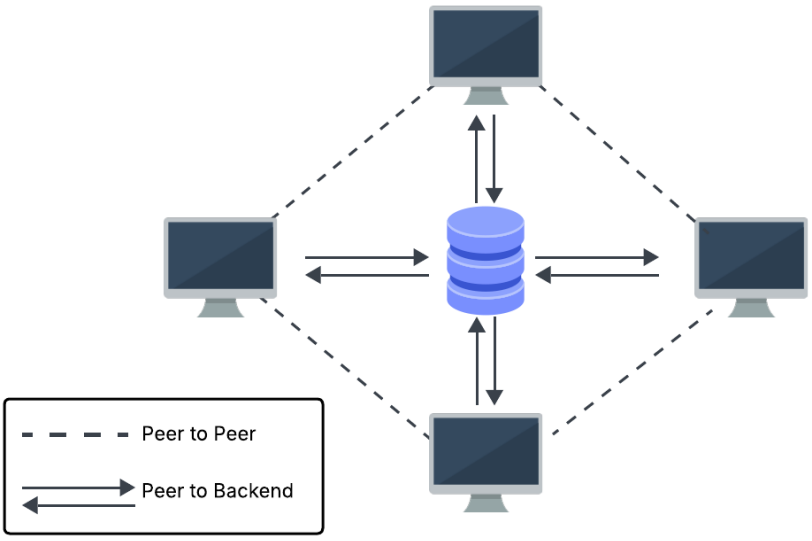


Figure 3.3: Network Architecture

3.2.2.1 Frontend Layer:

Presentation Layer represents the user interface of the system. This layer is responsible for interacting with end users. Its responsibilities:

- Facilitating dealing with system features.
- Interact with backend via REST API built with Express.js to store or fetch data as

users credentials and products details. See example below 3.2.1 :

3.2.1 REST API Example

```
// Setup express router
const apiRouter = express.Router();

// Store a product
const res = await apiRequest("POST", "/api/products", data );

// Fetch user data
const res = await apiRequest("GET", "/api/user");
```

- Interact with blockchain network by listening for messages.

3.2.2.2 Backend Layer:

The backend acts as a mediator between the frontend and the database. Its responsibilities:

- Handling API requests that come from the frontend, checks permissions, and validates data.
- Checking permissions based on user role that retrived from token before allowing any sensitive operations. Role permission check 3.2.2:

3.2.2 Checking if role has permission

```
export function roleMiddleware(allowedRoles: string[]) {
  return (req: Request, res: Response, next: NextFunction) => {
    if (!req.user || !allowedRoles.includes(req.user.role)) {
      return res.status(403).json({ message: "Access denied"
          });
    }
    next();
  };
}
```

- Interacting with postgresql database to store product data, and user profiles and roles.

3.2.2.3 Database Layer:

Relational database added to achieve performance, speed, scalability and security.

3.2.2.4 Blockchain Layer:

Network acts as the core of the system. It handles decentralized communication between clients (peers), enabling a fully distributed system for product tracking and blockchain updates.

3.2.2.5 Process Flow Design (3.4):

- **User Authentication:** Stakeholders authenticate using JWT tokens to access system functionalities.
- **Product Registration:** Authorized users register new pharmaceutical products with detailed metadata.
- **Transfer Initiation:** Product transfers between supply chain participants trigger blockchain transactions.
- **Real-time Notification:** WebSocket and WebRTC connections broadcast transaction events to all connected participants.
- **Mining Process:** Administrative nodes receive notifications and initiate the mining process for new blocks.
- **Consensus Achievement:** The first admin to complete mining broadcasts the new block to the network.
- **Block Validation:** All network participants validate and append the new block to their local blockchain copy.

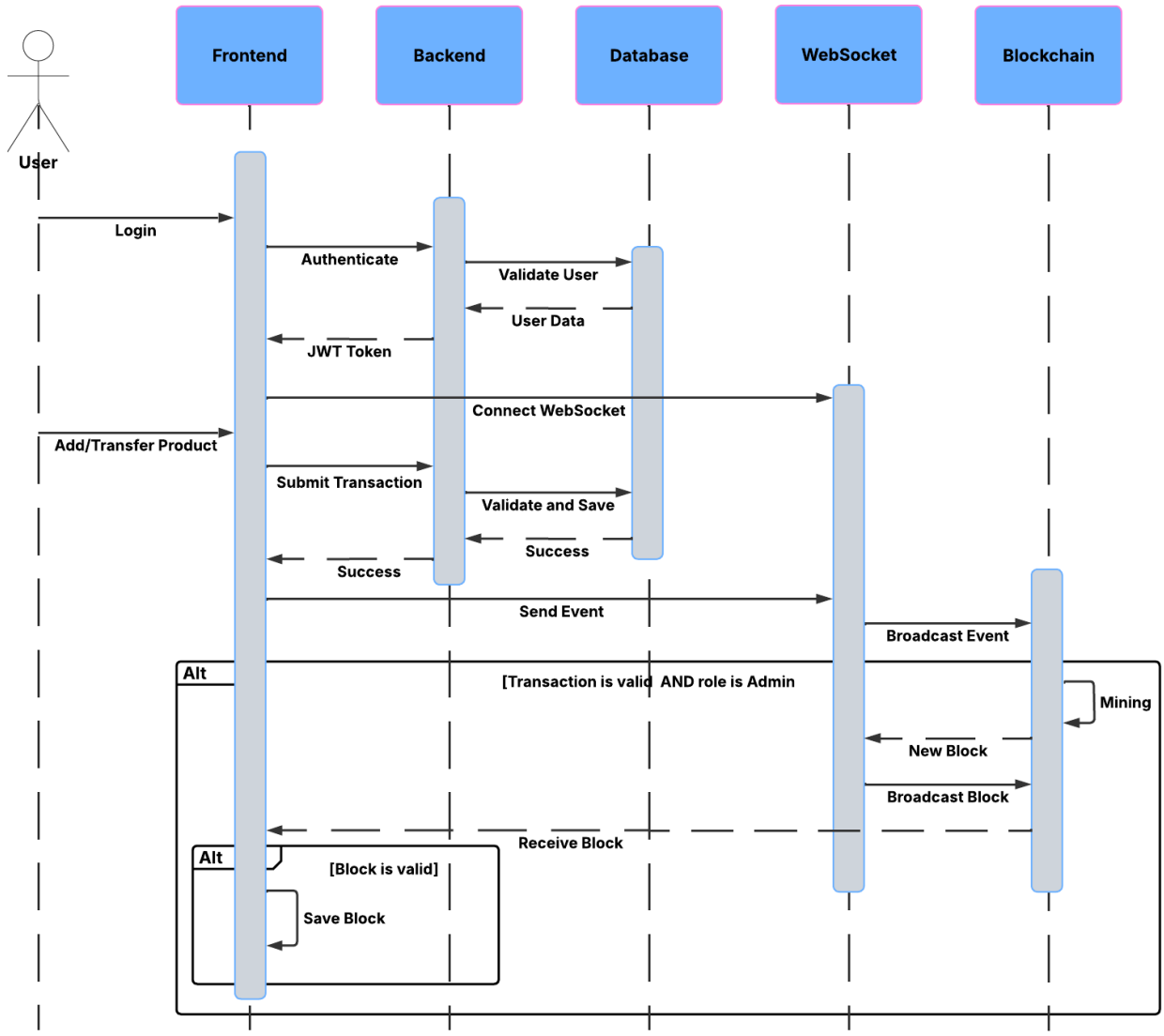


Figure 3.4: Process Flow Design

3.3 Implementation:

3.3.1 Technologies used for implementation:

The system was implemented on github codespace (2-core, 8GB RAM, 32GB storage capacity), and data stored in Neon Database.

The system utilizes modern web technologies and tools for frontend and backend development, database integration, and blockchain operations.

3.3.1.1 Frontend:

The user interface was built using:

- **React.js:** A declarative JavaScript library used for building responsive and dy-

namic user interfaces. Components like product registration forms, dashboards, and blockchain viewers were developed as reusable functional components.

- **TypeScript:** TypeScript is a strongly typed programming language that builds on JavaScript[19], it helps prevent runtime errors and improves maintainability.
- **Tailwind CSS:** A utility-first CSS framework that accelerates the development of modern UI layouts. It allows for rapid customization without writing custom CSS.[20]
- **Radix UI:** Provides unstyled but accessible UI primitives.[21] It ensures WAI-ARIA-compliant components like modal dialogs, dropdowns, and tabs used in the dashboard.

3.3.1.2 Backend:

The backend was developed using:

- **Node.js:** Node.js is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.[22] It enables asynchronous request handling and event-driven architecture.
- **Express.js:** Express is a minimal and flexible web application framework[23] that used to structure the API endpoints (e.g., /api/product/register) and middleware for validation and authorization..
- **JSON Web Token:** A JSON Web Token (JWT) is a compact, URL-safe way of representing claims between two parties.[24] Used for securing user authentication. Tokens are issued upon login and validated for protected actions like adding products or transferring them.

3.3.1.3 Database:

The database was developed using:

- **PostgreSQL:** PostgreSQL is a powerful, open source object-relational database.[25] Used to store user accounts, product details, and blockchain metadata.
- **Drizzle ORM:** Drizzle ORM is a TypeScript ORM for SQL databases designed with maximum type safety in mind.[26] Used to store user accounts, product details, and blockchain metadata.

3.3.1.4 Blockchain:

The blockchain was developed using those libraries:

- **Simple-Peer:** A JavaScript library enables you to create peer-to-peer connections in a web browser. It establishes a connection between two or more browsers using WebRTC (Web Real-Time Communication), enabling direct data exchange without the involvement of a server.[27]
- **Socket.IO:** Socket.IO is a library that enables low-latency, bidirectional and event-based communication between a client and a server.[28] Used to Handle signaling between peers to initiate WebRTC connections.
- **Crypto:** Cryptography library, used to compute SHA-256 hashes for each block during mining.

3.3.2 Database Implementation:

We used PostgreSQL as the database management system for its reliability, support for complex relational structures, and compatibility with standard SQL. The interaction with the database is handled using the Drizzle ORM, which integrates seamlessly with our TypeScript backend.

We create only two tables, Users and Products tables, where each user can create one or more product.

3.3.2.1 Detailed Tables Schema:

The Users table stores user information as Shown in Table (3.1).

Table 3.1: Users Table in the Database

Column	Data Type	Constraints	Default Value	Description
id	SERIAL	PRIMARY KEY	Auto-increment	Unique identifier for each user
username	TEXT	UNIQUE, NOT NULL	-	User's unique login identifier
password	TEXT	NOT NULL	-	Encrypted user password
role	TEXT	NOT NULL	'RegularUser'	User's system role/permissions

The Products table contains product details as Shown in Table (3.2).

Table 3.2: Products Table in the Database

Column	Data Type	Constraints	Default	Description
id	SERIAL	PRIMARY KEY	Auto-increment	Unique identifier
serial_id	TEXT	UNIQUE, NOT NULL	-	Product's serial number
name	TEXT	NOT NULL	-	Product name/title
manufacturer	TEXT	NOT NULL	-	Manufacturing company
manufacturing_date	DATE	NOT NULL	-	Date manufactured
expiration_date	DATE	NOT NULL	-	Product expiration date
created_at	TIMESTAMP	NOT NULL	now()	Record creation timestamp

3.3.2.2 Entity Relationship Diagram (3.5):

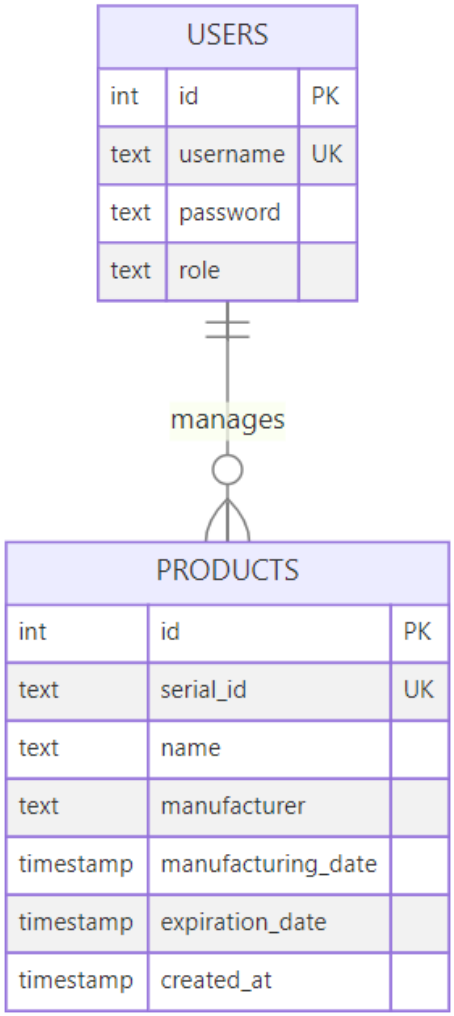


Figure 3.5: Entity Relationship Diagram

3.3.3 Blockchain Implementation:

A custom blockchain module was implemented using TypeScript and three main libraries (Simple-Peer, Socket.IO, and Crypto).

Implementation divided into two major functional part:

3.3.3.1 Network Initialization:

By using Socket.IO, we initialized a socket in frontend as shown in code 3.3.1:

3.3.1 Initialization of Socket

```
// connecting to a signaling server
const socket =
   io('https://urban-yodel-x545vxw999qw3pg5j-5000.app.github.dev/',
   {
  transports: ['websocket'], // Optional but recommended
});

const peerConnections: Record<string, ReturnType<typeof
   createPeer>> = {};

// notify the signaling server new user
export function initializeSignaling(userId: string) {
  socket.emit('register', userId);
}
```

And we impelemented a signaling server in backend as shown in code 3.3.2:

3.3.2 Signaling Server Code

```
  // registers new user
  socket.on('register', (userId) => {
    connectedUsers.set(userId, socket.id);
    socket.join(userId);
  });

 // handles signaling messages
  socket.on('signal', ({ to, from, signal }) => {
    io.to(to).emit('signal', { from, signal });
  });

 // delete a user when he disconnects
  socket.on('disconnect', () => {
    for (const [userId, sId] of connectedUsers.entries()) {
      if (sId === socket.id) {
```

```
16          connectedUsers.delete(userId);
17          break;
18        }
19      }
20    });
```

3.3.3.2 Network Working:

After network initialization and users forming a peer-to-peer network, network start work like blockchain mechanism.

- When a user create or transfer a product, -handleNewProduct- function or -handleTransferProdu function is called to initialize a message and broadcast it. As shown in code below 3.3.3.

3.3.3 Handle New Prodcut

```
1  export function handleNewProduct(product: ProductData, user: {
       id: string, role: UserRole, name: string }) {
2    const message: PeerMessage = {
3      type: 'NEW_PRODUCT',
4      product,
5      from: user.id,
6      role: user.role,
7      name: user.name,
8    };
9    broadcastMessage(message);
10 }
11
12 export function handleTransferProduct(product: ProductData,
       user: { id: string, role: UserRole , name: string }) {
13   const message: PeerMessage = {
14     type: 'TRANSFER_PRODUCT',
15     product,
16     from: user.id,
17     role: user.role,
18     name: user.name,
19   };
20   broadcastMessage(message);
21 }
```

- Message broadcasted with 3.3.4:

3.3.4 Broadcast Message

```
export function broadcastMessage(msg: PeerMessage) {
  for (const peer of Object.values(activePeers)) {
    if ((peer.dataChannel as Peer.Instance).connected) {
      peer.dataChannel.send(JSON.stringify(msg));
    }
  }
}
```

- When users receive a message they call -handleIncomingMessage- as shown in code 3.3.5, that will let only users with admin role see products messages. Block messages will automatically validates and saves for all nodes.

3.3.5 Handle Incoming Message

```
export function handleIncomingMessage(raw: string, user: {
    role: UserRole, id: string }) {
  const message: PeerMessage = JSON.parse(raw);

  if ((message.type === 'NEW_PRODUCT' || message.type ===
      'TRANSFER_PRODUCT') && user.role === 'Admin') {
    // Notify the UI to display the product data
    if (onProductMessage) {
      onProductMessage(message.type, message.product,
          message.from, message.name);
    }
  } else if (message.type === 'NEW_BLOCK') {
    // Automatically validate and save the block for all peers
    const lastBlock = getLastBlock();
    const block = message.block;
    if (isValidBlock(block, lastBlock)) {
      saveBlockToLocal(block);
  }
}
```

- After that, a block is initialized as Shown in code 3.3.6. Every block is tied with his previous block with previous block hash.

3.3.6 Block Initialization

```
 const lastBlock = getLastBlock();
  const blockType = type === 'NEW_PRODUCT' ? 'Creation' :
      'Transfer';
  const block: Block = {
```

```
    index: lastBlock.index + 1,
    type: blockType,
    timestamp: Date.now(),
    data: JSON.stringify(product),
    previousHash: lastBlock.hash,
    nonce: 0,
    hash: '',
    currentOwner: name
  };
```

- Getting last block with this function 3.3.7. The function create genesis block if the blockchain is empty.

3.3.7 Fetch Last Block

```
export function getLastBlock(): Block {
  let currentChain = loadBlockchainFromLocal();
  if (currentChain.length === 0) {
    const genesisData: ProductData = {
      name: "Genesis Product",
      serialId: "",
      manufacturer: "",
      manufacturingDate: "",
      expirationDate: "",
      createdAt: "",
    };
    const genesisBlock: Block = {
      index: 0,
      type: "Genesis",
      timestamp: Date.now(),
      data: JSON.stringify(genesisData),
      previousHash: "0",
      nonce: 0,
      hash: "0000000000000000",
      currentOwner: "",
    };
    const updatedChain = [...currentChain, genesisBlock];
    localStorage.setItem(BLOCKCHAIN_KEY,
       JSON.stringify(updatedChain));
    currentChain = loadBlockchainFromLocal();

  }
  const lastBlock = currentChain[currentChain.length - 1];
  return lastBlock;
}
```

- Then -mineBlock- function 3.3.8 is called to mine the block with PoW.

3.3.8 Block Mining

```
function mineBlock(block: Block): Block {
  const previousBlock = getLastBlock();
  let nonce = 0;
  let hash = '';
  const difficulty = '0000';
  const index = block.index ;
  const previousHash= previousBlock.hash;
  const timestamp= block.timestamp;
  while (true) {
    hash = calculateHash(
      index.toString() +
      timestamp.toString() +
      block.data +
      previousHash +
      nonce.toString(),
    );
    if (hash.startsWith(difficulty)) break;
    nonce++;
  }
  block.hash = hash;
  block.nonce = nonce;
  return {
    ...block
  };
}
```

- The -hashCalculate- function 3.3.9 that used in the mine of the block, that use SHA256.

3.3.9 Calculate Hash

```
export function calculateHash(data string) string {
    return crypto.createHash(sha256).update(data).digest(hex);
  }
```

- A message of type "New Block" is broadcated after a block is mined as shown in code 3.3.10.

3.3.10 Broadcast New Block

```
const newMessage: PeerMessage = {
    type: 'NEW_BLOCK',
    block: minedBlock
  };
broadcastMessage(newMessage);
```

- Users check if the new block is valid with the below function 3.3.11.

3.3.11 Block Validation

```
function isValidBlock(newBlock: Block, previousBlock: Block):
   boolean {
  if (newBlock.previousHash !== previousBlock.hash) return
     false;
  if (newBlock.index !== previousBlock.index + 1) return false;
  return true;
}
```

- Users save the new valid block in their local blockchains with the below function 3.3.12.

3.3.12 Block Saving

```
const BLOCKCHAIN_KEY = "peer_blockchain";

export function saveBlockToLocal(newBlock: Block) {
  const currentChain = loadBlockchainFromLocal();
  const updatedChain = [...currentChain, newBlock];
  localStorage.setItem(BLOCKCHAIN_KEY,
     JSON.stringify(updatedChain));
}
```

- The function that load the blockchain 3.3.13.

3.3.13 Load Blockchain

```
export function loadBlockchainFromLocal(): Block[] {
  const data = localStorage.getItem(BLOCKCHAIN_KEY);
  return data ? JSON.parse(data) : [];
}
```

- Search function 3.3.14 that return prodcut history.

3.3.14 Search Product

```
export function searchBlocksBySerialId(serialId: string):
   Block[] {
  const blockchain = loadBlockchainFromLocal();
  const result: Block[] = [];

  for (const block of blockchain) {
    try {
      const data: ProductData = JSON.parse(block.data);
      if (data.serialId === serialId) {
        result.push(block);
      }
    } catch (err) {
      continue;
    }
  }

  return result;
}
```

Implementation Summary Table

Component	Technologies Used
Environment	GitHub Codespaces (2-core CPU, 8GB RAM, 32GB storage), Neon Database
Frontend	React.js (UI components), TypeScript (type safety), Tailwind CSS (UI styling), Radix UI (accessible primitives)
Backend	Node.js (runtime), Express.js (API routing), JSON Web Token (authentication)
Database	PostgreSQL (relational DB), Drizzle ORM (TypeScript ORM for queries and type safety)
Blockchain Layer	Simple-Peer (P2P via WebRTC), Socket.IO (signaling and communication), Crypto (SHA-256 hashing for block mining)

Table 3.3: Overview of technologies used in the system implementation

3.3.4 Tamper-Proofing and Chain Integrity:

3.3.4.1 Role of Cryptographic Hashing in Chain Binding:

Every block is tied with previous block with a hash making a chain. If any data in a block is altered even a single character the computed hash will change entirely, and this will broke the chain. The chain is validated every time user want to display it, if the user altered some data the chain will not display and he will get an error.

3.3.4.2 Tampering Isolation: Individual Chains are Local:

Decentralized peer-to-peer setup ensure that each peer have its own local copy of the blockchain. If a user modifies his local chain, other blockchain of other users not effected with this modification.

3.3.4.3 Daily Chain Synchronization and Consensus Enforcement:

Users who changed their blockchain on propose or accidentally are obligated to synchronized with other users.

All users automatically apply this built-in synchronization mechanism daily:

- Each peer broadcasts their full blockchain to others.
- Each peer collects all received chains and groups them by their cryptographic digest.
- The version shared by the most peers is the correct version (compare only valid chains).
- Peers with different chains replace their version with the majority-consensus chain as shown in code 3.3.15.

3.3.15 Sync Mechanism

```
1  function resolveChainConsensus() {
2    const digestMap: Record<string, Block[]> = {};
3    const counts: Record<string, number> = {};
4
5    for (const chain of collectedChains) {
6      const digest = getChainDigest(chain);
7      if (!digestMap[digest]) digestMap[digest] = chain;
8      counts[digest] = (counts[digest] || 0) + 1;
9    }
10
11   const [majorityDigest, count] = Object.entries(counts).sort((a,
       b) => b[1] - a[1])[0];
12
```

```
13    const currentDigest = getChainDigest(loadBlockchainFromLocal());
14
15    if (majorityDigest !== currentDigest) {
16      const majorityChain = digestMap[majorityDigest];
17      localStorage.setItem(BLOCKCHAIN_KEY,
            JSON.stringify(majorityChain));
18    }
19  }
```

3.4 Experimental Process:

This section describes the experimental process conducted to validate the usability and functionality of the blockchain-based product tracking system created. The test environment, input data, and results obtained are provided with the key functions of the system.

3.4.1 Test Environment:

The application was deployed and tested in a GitHub Codespaces environment configured with 2 vCPUs, 8 GB RAM, and 32 GB of storage, running on Ubuntu 22.04, lunched in Opera, Edge, and Chrome browsers.

3.4.2 Data Used:

Test data are sample products available in Algeria with attributes such as name, manufacturer, manufacturing date, and expiration date (the serial id is not real). Simulated transfer paths were defined between manufacturers, distributors, and pharmacies.

3.4.3 Obtained Results:

The system successfully demonstrated its functional capabilities. All user interactions, including product creation, transfer, and traceability checks, were recorded in the blockchain without data inconsistency. Figures below illustrate the main system features validated during the testing phase.

3.4.4 Functional Overview:

The system offers the following features:

- **Sign-In Page:** A login interface for users and admins to securely access the system as shown in Figure (3.6).

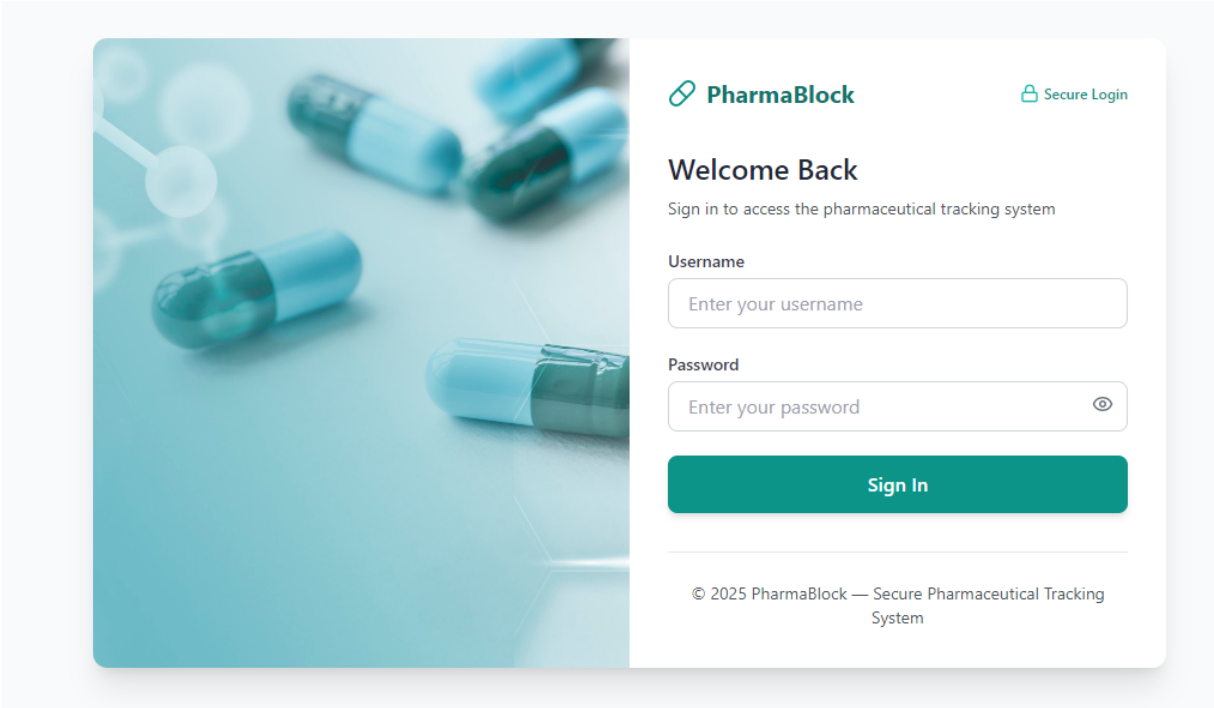


Figure 3.6: Sign in Page

- **User Dashboard:** Users can view Total Products, Total Transfers, Blockchain Size, and Recent Activity as shown in Figure (3.7).

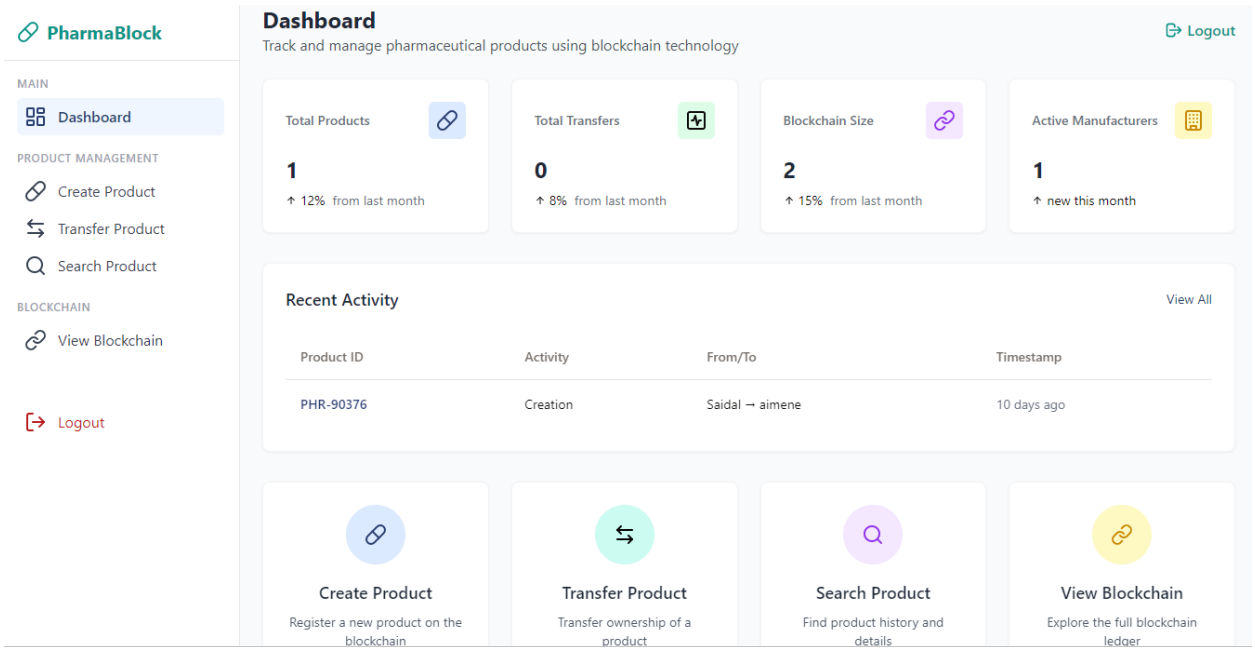


Figure 3.7: Dashboard Page

- **Product Registration:** The Manufacturer registers a product with key data (name, manufacturer, expiration date, etc.). In this example we registers two real products. as shown in Figures (3.8), (3.9).

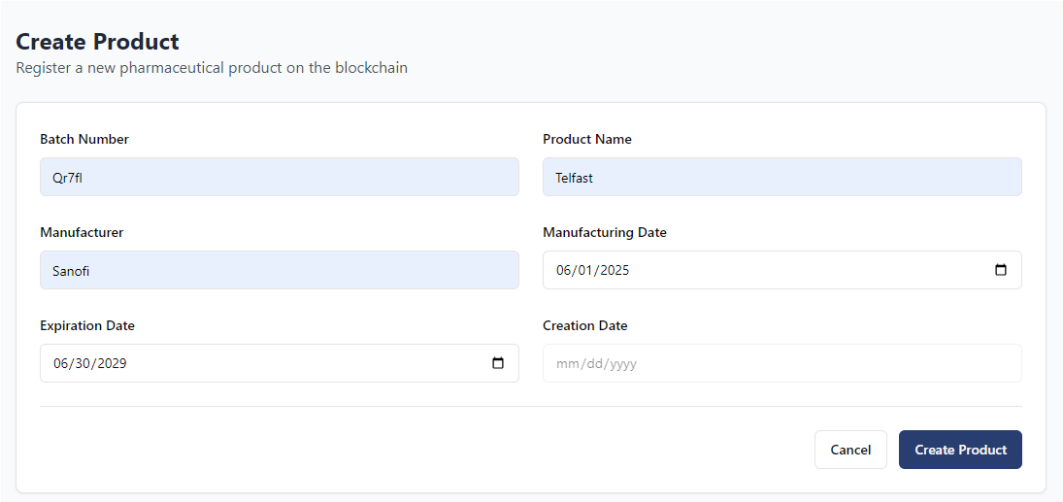


Figure 3.8: Create Product 1

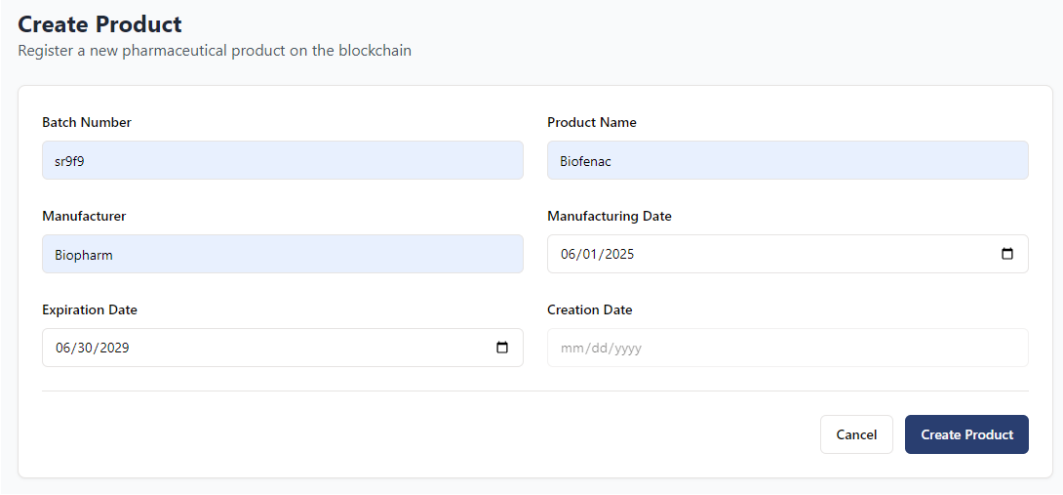


Figure 3.9: Create Product 2

- **Product Transfer:** Products are transported from the factory to the pharmacy, passing through several stations, with each movement recorded as a block. The Example in the shown figures (3.10), (3.11) shows transfer.

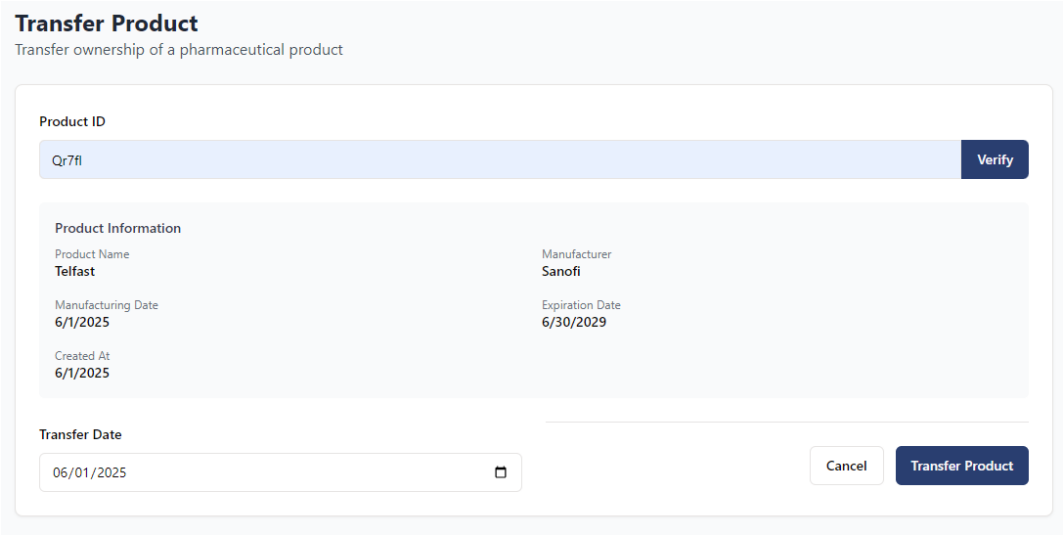


Figure 3.10: Transfer Product

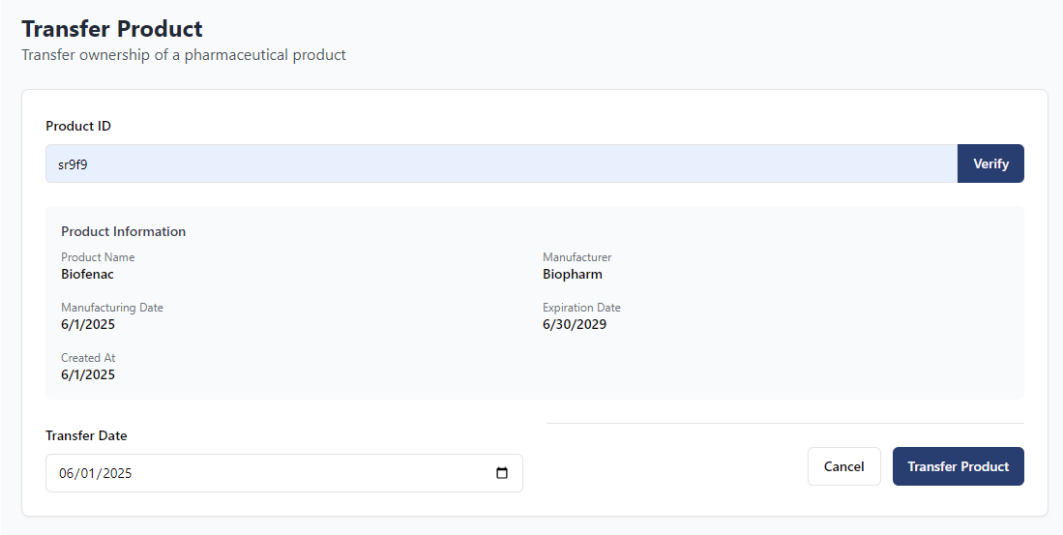


Figure 3.11: Transfer Product

- **Product Mining:** Once a product registration or transfer lunched it shows directly on admin dashboard as shown in figure (3.12) to mine.

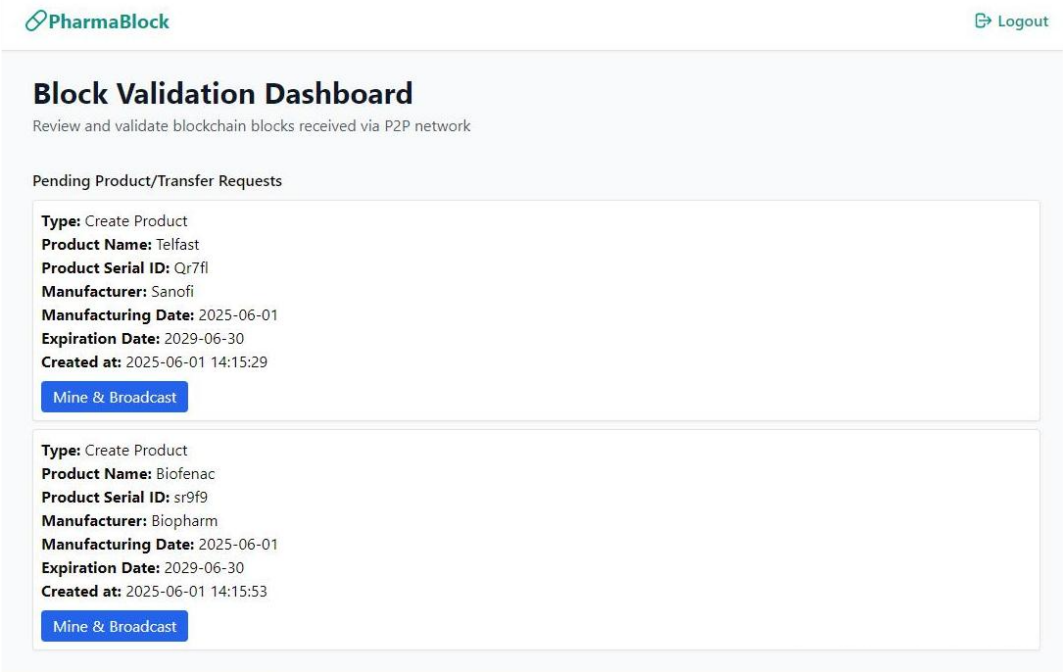


Figure 3.12: Admin Dashboard

- **Product History:** Users can search for a product by his Serial ID to see its journey as shown in Figure (3.13).

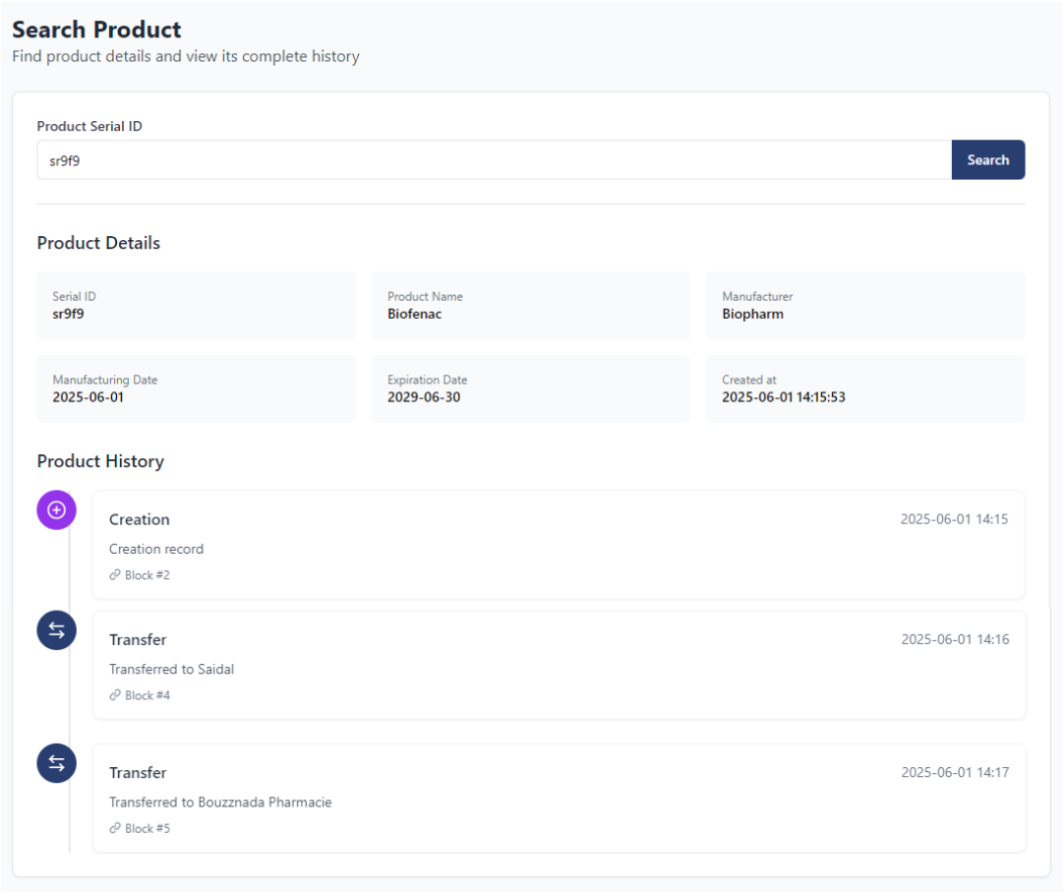


Figure 3.13: Search Page

- **Product Statistics:** Users can search for a product by his name to see his statistics as shown in Figure (3.14).

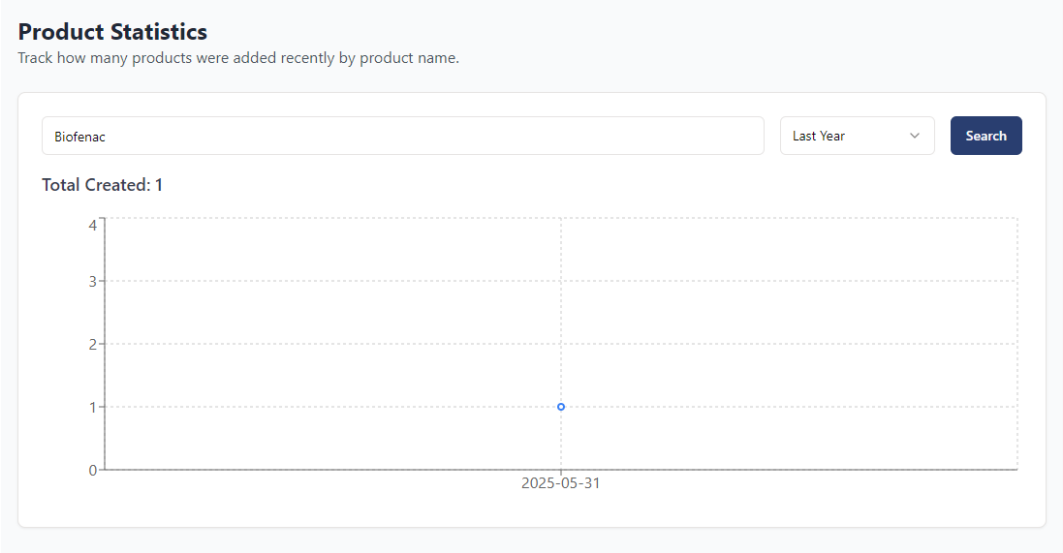


Figure 3.14: Product Statistics Page

- **Blockchain Preview:** Users can visualize the blockchain block by block and verify if it's valid or not as shown in Figure (3.15).

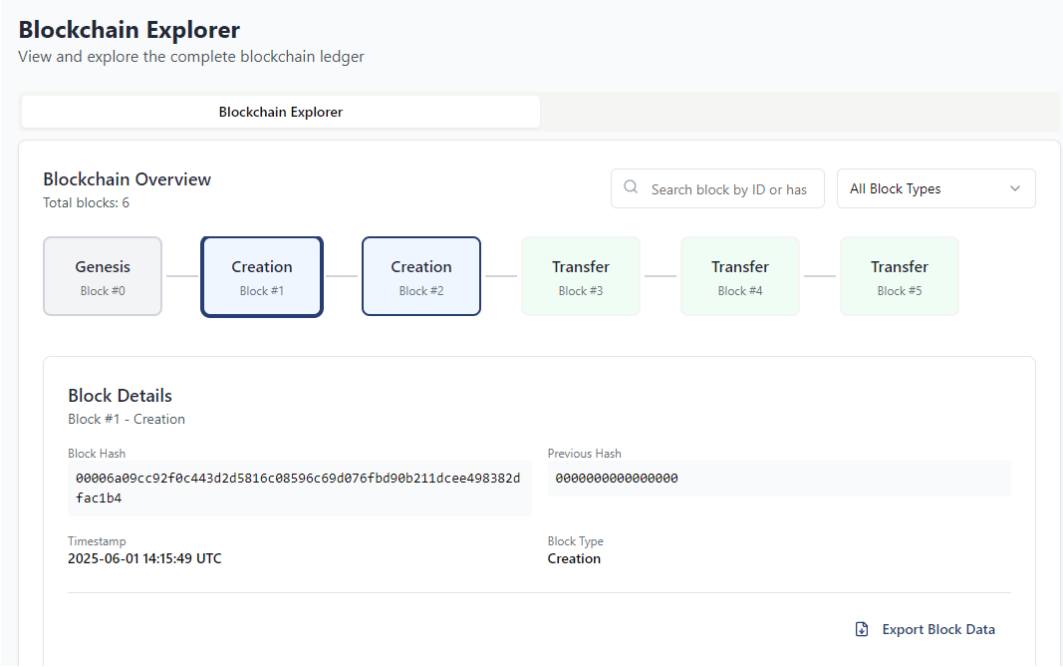


Figure 3.15: Blockchain Visualization Page

3.5 Conclusion:

In this chapter, we have presented the key contributions of our project, which aims to enhance pharmaceutical product traceability using a blockchain-based system. We began by outlining the theoretical framework, including the project objectives, system design and architecture.

The implementation phase demonstrated how modern web technologies—such as React, Node.js, PostgreSQL, and a custom-built blockchain—can be integrated to create a robust and user-friendly traceability platform. Each system component was carefully developed and tested to ensure that transactions are securely recorded and verifiable across the supply chain.

General Conclusion

Conclusion:

A revolutionary step in resolving persistent issues with traceability, transparency, efficiency, and trust is the incorporation of blockchain technology into global supply chain management. A decentralized and unchangeable ledger system, blockchain provides strong solutions for improving data integrity and visibility in a variety of supply chain operations.

Strong promise for scalable and secure supply chain solutions is shown by the quick development of blockchain platforms. A promise future where supply chains may function more independently, safely, and transparently is indicated by emerging trends like blockchain and IoTs.

To improve blockchain models, create global standards, and create a legal framework that encourages innovation, more study and cooperation between academic institutions. business, and legislators will be necessary. Lastly, blockchain is a key component of supply chain management's digital transformation, offering resilience and adaptability in an increasingly complex and interconnected world.

Bibliography

[1] echoroukonline: Over One and A Half Million Counterfeit Medicines Reach Algeria. (2016)

[2] https://www.pfizer.com/products/medicine-safety/counterfeiting

[3] statista.com/chart/30067/worldwide-counterfeit-pharmaceuticals-incidents/

[4] Boppana, V.: Blockchain Applications in CRM for Supply Chain Management. (2024)

[5] Yang, L.: The blockchain: State-of-the-art and research challenges. (2019)

[6] Habib, G., Sharma, S., Ibrahim, S., Ahmad, I., Qureshi, S., Ishfaq, M.: Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. (2022)

[7] https://bitcoin.org/en/full-node#secure-your-wallet

[8] https://pharmasource.global/content/pharma-supply-chain-a-comprehensive-guide-to-the-pharmaceutical-supply-chain/

[9] https://www.techtarget.com/pharmalifesciences/feature/Fundamentals-of-the-Pharmaceutical-Supply-Chain

[10] Shivani, Z., Aditi, S.: Pharmaceutical Market Size, Trends, Top Companies and Strategies by 2034. (2025)

[11] Ostwald, D., Cramer, M., Albu, N., Tesch, J.: The Global Economic Impact of the Pharmaceutical Industry. (2020)

[12] Musamih, A., Salah, K., Jayaraman, R., Arshad, J., Debe, M., Al-Hammadi, Y., Ellahham, S.: A Blockchain-Based Approach for Drug Traceability in Healthcare Supply Chain. (2021)

[13] Mani, V., Prakash, M., Lai, W. C.: Cloud-based blockchain technology to identify counterfeits. (2022) https://www.ncbi.nlm.nih.gov

[14] Bandhu, C. K., Litoriya, R., Lowanshi, P., Jindal, M., et al.: Making drug supply chain secure, traceable, and efficient: a Blockchain and smart contract-based implementation. (2022) https://www.ncbi.nlm.nih.gov

[15] Sezer, B., Topal, S., Nuriyev, U.: An Auditability, Transparent, and Privacy-Preserving for Supply Chain Traceability Based on Blockchain. (2021)

[16] Sim, C., Zhang, H., Chang, M. L.: Improving End-to-End Traceability and Pharma Supply Chain Resilience using Blockchain. (2022) https://www.ncbi.nlm.nih.gov

[17] Zoughalian, K., Marchang, J., Ghita, B.: A Blockchain Secured Pharmaceutical Distribution System to Fight Counterfeiting. (2022) https://www.ncbi.nlm.nih.gov

[18] https://react.dev

[19] https://www.typescriptlang.org

[20] https://tailwindcss.com

[21] https://www.radix-ui.com

[22] https://nodejs.org/

[23] https://expressjs.com

[24] https://jwt.io/introduction

[25] https://www.postgresql.org/about

[26] https://supabase.com/docs/guides/database/drizzle

[27] https://medium.com/stackanatomy/introduction-to-simple-peer-a-webrtc-library-ab04ea8aa5fe

[28] https://socket.io/docs/v4/