



# **Mekelle University**

Ethiopian Institute of Technology – Mekelle (EiT-M)

Faculty of Computing

## **Integrating Ethereum Blockchain and GraphQL for a Secure Graduate Verification System at Mekelle University**

By Tilahun Mamuye Gidey

ID. Number: EiT/PR181425/13

Program: MSc. In Software Engineering, Regular

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science  
in Software Engineering.

Advisors

Behailu Getachew (PhD)

Co-Advisor

Destalem Negusse (MSc)

Mekelle University

Ethiopian Institute of Technology-Mekelle (EiT-M)

Mekelle, Ethiopia

Nov , 2025

Declaration

I, Tilahun Mamuye Gidey, declare that this thesis is my original work and has not been submitted elsewhere for any academic award. All sources of information have been properly acknowledged.

Signature:  \_\_\_\_\_

Date: January 2026

Approval Page

This is to certify that the thesis titled “**Integrating Ethereum Blockchain and GraphQL for a Secure Graduate Verification System at Mekelle University.**” Submitted by Tilahun Mamuye Gidey in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering, has been examined and approved.

Supervisor: Behailu Getachew (PhD)

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Department Head: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

# Mekelle University

Ethiopian Institute of Technology – Mekelle (EiT-M)

Faculty of Computing

## “Integrating Ethereum Blockchain and GraphQL for a Secure Graduate Verification System at Mekelle University”

Tilahun Mamuye Gidey

Program: MSc in Software Engineering, Regular.

Approved By:

Examining Boards:

1. Mehamed Ahmed(PhD) : \_\_\_\_\_

2. Guesh Dagneu (PhD) : \_\_\_\_\_

3. Behailu Getachew (PhD) : \_\_\_\_\_

4. Selama Gebremeskel (Chair Person)  *Selama Gebremeskel  
Jan 26, 2026*

## **Abstract**

Ensuring the integrity and efficiency of academic record verification has become increasingly important for modern educational institutions. This study presents a blockchain-powered verification system specifically designed for confirming the credentials of graduated students from Mekelle University. By integrating Ethereum blockchain with GraphQL APIs, the system enhances transparency and reliability in the verification process. The university's existing system, built with Ruby on Rails, lacked automated verification, relied heavily on centralized control, and was prone to delays and potential data tampering.

To overcome these issues, a decentralized application (DApp) was developed using various tools, including Ethers.js, Node.js, Ganache, Apollo Server, GraphQL, and React. This application enables the secure submission and retrieval of student records through Ethereum smart contracts. Data can be uploaded via CSV files or manually entered through forms, and each record is retrievable using a unique student ID, ensuring data immutability and public verifiability.

Stakeholder feedback was gathered through interviews, and thematic analysis was used to assess the system's usability, scalability, and trustworthiness. Findings showed strong support for the blockchain-based system, with over 90% of participants agreeing that it improves transparency and reduces the risk of credential fraud.

This research demonstrates a feasible bridge between traditional university information systems and decentralized technologies, highlighting both the practicality and institutional readiness for adopting blockchain in higher education.

**Keywords:** Blockchain, Smart Contracts, GraphQL, Student Verification, Ethereum, DApp, Mekelle University, Academic Records

## Dedication

To my family, who bring joy and inspiration to my life every day.



## **Acknowledgments**

First and foremost, I would like to thank Almighty God for giving me the opportunity and strength to complete this thesis with perseverance and success. I am also grateful for the support and encouragement of my family during the specialization. I would also like to express my sincere gratitude to my Advisor, Dr. Behailu Getachew, for his continuous supervision of my research and for his enthusiasm and great knowledge, which have greatly helped me to complete my research. I would like to express my deepest gratitude to my co-advisor, Mr. Destalem Negusse, for his great help with my thesis work. I would also like to express my deepest appreciation to the School of Computing at Mekelle University for the opportunities it has provided me to pursue my research interests.

## **Table of Contents**

|   |      |
|---|------|
| List of Figures .....   | xi   |
| List of Tables .....  | xii  |
| List of Acronyms .....  | xiii |
| Chapter 1 : Introduction .....                                      | 1    |
| 1.1 Background of the Study .....                                   | 1    |
| 1.2 Motivation .....  | 2    |
| 1.3 Statement of the Problem .....                                  | 3    |
| 1.4 Research Questions .....  | 4    |
| 1.5 Research Objectives .....                                       | 5    |
| 1.5.1 Major Objective .....   | 5    |
| 1.5.2 Specific Objectives .....                                     | 5    |
| 1.6 Scope of the Research .....                                     | 5    |
| 1.7 Significance of the study .....                                 | 6    |
| 1.8 Contribution of the study .....                                 | 7    |
| 1.9 Thesis structure.....   | 7    |
| Chapter 2 : Literature Review.....                                  | 8    |
| 2.1 Introduction .....  | 8    |
| 2.2 Blockchain Technology in Academic Credential Verification ..... | 8    |
| 2.3 Smart Contracts for Secure Academic Records .....               | 8    |
| 2.4 Decentralization Theory .....                                   | 9    |
| 2.5 Data Integrity and Immutability .....                           | 9    |
| 2.6 GraphQL for Data Access and Efficiency.....                     | 9    |
| 2.7 Existing Student Verification Systems.....                      | 10   |
| 2.8 Web 3.0 in Blockchain-Based Systems.....                        | 10   |
| 2.9 Blockchain Frameworks for Academic Verification Systems .....   | 11   |
| 2.10 Blockchain Security Mechanisms .....                           | 11   |

|   |        |
|---|--------|
| 2.11 Cryptocurrency Wallets and Gas Fees .....                  | 12     |
| 2.12 Smart Contract Programming Languages .....                 | 12     |
| 2.13 Research Gap Analysis.....                                 | 13     |
| 2.14 Research Gap.....  | 13     |
| 2.15 Theoretical Framework .....                                | 16     |
| 2.16 Conceptual framework .....                                 | 17     |
| 2.17 General Data Protection Regulation (GDPR) .....            | 18     |
| <br>Chapter 3 : Methodology .....                               | <br>20 |
| 3.1 System Design and Architecture .....                        | 20     |
| 3.2 Smart Contract Development .....                            | 20     |
| 3.3 Smart Contract Platform Selection.....                      | 21     |
| 3.3.1 Qualitative Comparison of Blockchain Platforms .....      | 21     |
| 3.3.2 Quantitative Comparison and Empirical Justification ..... | 21     |
| 3.3.3 Final Justification for Choosing Ethereum .....           | 22     |
| 3.4 Data Integration and Upload .....                           | 22     |
| 3.5 GraphQL API Implementation.....                             | 22     |
| 3.6 Frontend Development .....                                  | 23     |
| 3.7 Testing and Validation .....                                | 23     |
| 3.8 Research Design Justification .....                         | 23     |
| 3.9 Constructivism in Educational Technology .....              | 23     |
| 3.10 Institutional Theory .....                                 | 23     |
| 3.11 Interview Data Analysis Method.....                        | 24     |
| 3.12 System analysis and design .....                           | 24     |
| 3.13 System Analysis .....                                      | 24     |
| 3.13.1 Existing System Overview.....                            | 24     |
| 3.13.2 Identified Problems .....                                | 24     |
| 3.13.3 Functional Requirements: .....                           | 25     |
| 3.13.4 Nonfunctional requirements:.....                         | 26     |
| 3.14 System Design.....   | 26     |
| 3.14.1 System Architecture.....                                 | 26     |

|  |    |
|--|----|
| 3.14.2 System Components and Workflow .....                      | 26 |
| 3.14.3 Data Flow Diagram .....                                   | 27 |
| 3.14.4 Smart Contract Design .....                               | 27 |
| 3.14.5 Security and Integrity Considerations .....               | 27 |
| 3.14.6 Design Advantages .....                                   | 28 |
| 3.14.7 Design Challenges.....                                    | 28 |
| Chapter 4 : Experimental setup and analysis .....                | 29 |
| 4.1 Technology Used.....   | 29 |
| 4.2 Architecture Used.....                                       | 30 |
| 4.3 Data Flow .....  | 31 |
| 4.4 Results .....  | 31 |
| 4.4.1 System Functionality.....                                  | 31 |
| 4.4.2 Comparison to the Existing System .....                    | 33 |
| 4.4.3 User Experience .....                                      | 33 |
| 4.4.4 Limitations .....  | 33 |
| 4.4.5 System Limitations and Data Integrity Considerations ..... | 34 |
| 4.5 Comparative Performance Evaluation.....                      | 34 |
| 4.6 Interview Data Analysis Approach .....                       | 35 |
| 4.6.1 Response Statistics and Analysis .....                     | 35 |
| 4.6.2 Summary Insights (with Statistics) .....                   | 35 |
| 4.6.3 Results and Discussion.....                                | 36 |
| 4.6.4 Key Statistics.....  | 37 |
| 4.6.5 Summary .....  | 37 |
| Chapter 5 : Conclusion and future work .....                     | 39 |
| 5.1 Discussion .....   | 39 |
| 5.2 Conclusion.....  | 39 |
| 5.3 Research Contributions .....                                 | 40 |
| 5.4 Future Work .....  | 40 |
| 5.5 Recommendation.....  | 41 |

|   |    |
|---|----|
| Reference .....   | 42 |
| Appendixes .....  | 45 |
| Appendix A: Interview Questions .....                                 | 45 |
| Appendix B: Smart Contract Security and Functional Audit Report ..... | 47 |
| Appendix C: Manual of the Developed Sample System .....               | 50 |
| Appendix D: Tools and Technologies Used .....                         | 52 |
| Appendix E: Basic codes of the pilot project implemented .....        | 53 |
| Appendix F: Interview Data .....                                      | 62 |

## List of Figures

|   |    |
|---|----|
| Figure 2.1 Conceptual framework of the system .....                                       | 18 |
| Figure 3.1 Dataflow diagram .....   | 27 |
| Figure 4.1: Local Ganache ETH Network .....   | 29 |
| Figure 4.2: Home page interface using React .....   | 30 |
| Figure 4.3: Successful Uploading CSV File. ....   | 31 |
| Figure 4.4: Successfully adding Student data .....  | 32 |
| Figure 4.5: Successful searching (Verifying) a student from the blockchain using ID ..... | 32 |

## List of Tables

|   |    |
|---|----|
| Table 2.1 List of related papers with their gap .....   | 16 |
| Table 3.1 Comparison between Ruby on Rails System and Blockchain-Based Student<br>Verification System ..... | 25 |
| Table 4.1: Stakeholders participated in the interview.....  | 35 |
| Table 4.2: Summary Insights (with Statistics).....  | 36 |
| Table 4.3 Thematic Summary.....   | 37 |

## List of Acronyms

|          |  |
|----------|--|
| API      | Application Program Interface  |
| CGPA     | Cumulative GPA   |
| CSV      | Comma-Separated Values   |
| DApp     | Decentralized Application  |
| ETH      | Etherium   |
| Gas Fee  | Transaction cost paid to execute operations on the Ethereum blockchain           |
| GDPR     | General Data Protection Regulation   |
| MVC      | Model-View-Controller  |
| RoR      | Ruby on Rails  |
| UI/UX    | User Interface/User Experience   |
| estudent | Electronic student management system of Mekelle University                       |
| SHA-266  | Secure Hash Algorithm 256-bit  |
| Gwei     | (Gigawei) is the standard unit for calculating gas fees on the Ethereum network. |
| PoS      | Proof-of-Stake   |

## **Chapter 1 : Introduction**

The rapid advancement of digital technologies has transformed how universities manage and verify academic credentials [1]. As institutions like Mekelle University adopt online platforms for student record management, ensuring data security, transparency, and trust has become a growing concern. Traditional centralized systems, such as Ruby on Rails, though efficient for internal data handling, face limitations in preventing tampering and supporting external verification [2]. This study introduces a blockchain-based verification framework that integrates Ethereum smart contracts with GraphQL APIs to enhance the authenticity, accessibility, and immutability of student records. By bridging decentralized blockchain technology with modern API design, the research aims to modernize Mekelle University's e-student system, enabling faster, more reliable, and transparent verification of graduates' academic credentials.

### **1.1 Background of the Study**

As educational institutions increasingly rely on digital systems, ensuring the security and integrity of student data has become a critical concern [3]. Technology providers play a key role in safeguarding these systems, yet they face growing challenges due to the rapidly evolving digital landscape, the increasing sophistication of cyber threats, and the complex dynamics between vendors and educational institutions [4].

At Mekelle University, the student grade management system is currently built using Ruby on Rails, a popular web application framework valued for its simplicity and productivity [5]. However, this centralized system poses several limitations, particularly in terms of security, data transparency, and trust among stakeholders. A centralized approach can make the system vulnerable to unauthorized access, manipulation, and operational delays, all of which are increasingly problematic in a world where data protection is paramount.

Given the rising demand for secure and transparent academic record systems, adopting modern technologies is essential. Blockchain technology, particularly smart contracts, offers a promising solution. Smart contracts are self-executing programs that enforce the terms of an

agreement through code, ensuring data immutability, trust, and automated execution without the need for intermediaries [6].

In addition, GraphQL, a powerful and flexible query language for APIs, enables more efficient, accurate, and customizable access to student data than traditional REST APIs [7]. Integrating GraphQL with blockchain can enhance data accessibility while preserving security.

This research proposes the integration of Ethereum-based smart contracts with GraphQL to build a decentralized student verification system tailored to Mekelle University's needs. Such a system would ensure greater transparency, enhanced data security, and a streamlined process for verifying graduates, especially important as cyberattacks become more frequent and institutions seek resilient technologies to manage sensitive academic records.

## **1.2 Motivation**

With the rise in global student mobility and the internationalization of higher education, cross-border verification of academic credentials has become increasingly common. A blockchain-based student information system enables institutions, embassies, and employers worldwide to verify academic records instantly, eliminating delays and uncertainties associated with manual verification processes. This aligns with Mekelle University's practical need to adopt global best practices in academic data management.

The growing demand for secure and tamper-proof verification of academic credentials poses a challenge for many institutions. While Mekelle University currently uses a Ruby on Rails-based Student Information Management System, it was primarily designed for internal and external use. As a result, it lacks transparency and security for efficient graduate verification, especially when requested by external bodies such as employers, international institutions, or government agencies.

These limitations serve as the driving force behind this research, which seeks to modernize Mekelle University's verification process through the integration of Smart Contracts and GraphQL into a decentralized system. Blockchain technology, particularly Ethereum smart

contracts, offers a secure and immutable method for recording student data. Once data is uploaded to the blockchain, it becomes tamper-resistant and trustworthy, eliminating concerns about data falsification [8].

GraphQL further complements this system by providing an efficient, flexible, and developer-friendly query interface for accessing blockchain data. With it, users can retrieve precise student records, such as by student ID, without the overhead of traditional API requests.

A functional prototype developed using Ethers.js, Node.js, Ganache, GraphQL, Apollo Server, and React demonstrates the technical viability of this approach. The system allows administrators to upload student data via CSV files or manual entry forms and enables seamless retrieval and verification of records stored on the blockchain.

Ultimately, this research is motivated by the vision of transitioning from Mekelle University's current centralized system to a decentralized, blockchain-based verification platform. The goal is to enhance trust, data integrity, and operational efficiency while aligning the university with international standards in academic verification.

### **Key Motivations for Integration:**

1. **Data Integrity and Security:** Blockchain ensures that student records are immutable and securely stored, minimizing the risk of tampering or unauthorized changes [9].
2. **Transparency and Trust:** Every transaction is recorded on a distributed ledger, creating a verifiable audit trail that fosters trust among stakeholders.
3. **Efficient Data Retrieval:** GraphQL allows users to request only the needed data, improving query performance and reducing overhead.
4. **Decentralization:** Removing reliance on centralized servers increases system resilience and eliminates a single point of failure.

### **1.3 Statement of the Problem**

The current student information system at Mekelle University, built using Ruby on Rails, operates on a centralized architecture that requires internal authority for data verification. This

centralization presents significant challenges, particularly for graduates seeking international education or employment, as third-party verification often involves delays and limited access.

Centralized systems are inherently vulnerable to risks such as unauthorized data access, manipulation, and single points of failure. Moreover, the inability of students, faculty, or administrators to independently verify changes to academic records reduces transparency and accountability. The lack of a tamper-resistant mechanism also compromises data integrity and increases exposure to security breaches.

To address these limitations, this research introduces a decentralized solution that integrates Ethereum smart contracts with GraphQL. This approach provides a secure, transparent, and efficient method for managing and verifying graduate records. Smart contracts eliminate the need for intermediaries, reduce verification delays, and enhance trust in the authenticity of data [10]. Meanwhile, GraphQL improves user experience and system responsiveness by enabling precise, client-controlled queries through a single endpoint, resulting in faster loading times and minimized data transfer [11].

By leveraging these technologies, the proposed system aims to transform Mekelle University's academic verification process, ensuring data immutability, improving stakeholder confidence, and supporting institutional modernization.

#### **1.4 Research Questions**

This study is guided by the following research questions:

- How can Smart Contracts be used to store and access the academic records of graduated students at Mekelle University without affecting the existing e-student?
- How can GraphQL be effectively integrated with Smart Contracts?
- What are the benefits of integrating Smart Contracts and GraphQL in terms of system transparency, security, and performance for Mekelle University's graduate verification process?

- What potential challenges may arise during the implementation of this system, and how can they be addressed?

## **1.5 Research Objectives**

### **1.5.1 Major Objective**

The major objective of this research is to design and implement a decentralized student verification system for Mekelle University by integrating Ethereum-based smart contracts with GraphQL APIs, in order to enhance the security, transparency, accessibility, and trustworthiness of student academic records.

### **1.5.2 Specific Objectives**

Specifically, this research seeks to:

1. Design and implement a smart contract capable of storing and managing student graduation records on the Ethereum blockchain.
2. Develop a GraphQL API using Apollo Server to allow flexible, efficient, and secure queries to access blockchain-based student data.
3. Build a React-based front-end interface to facilitate data input via CSV upload or form and allow third parties to verify credentials using student IDs.
4. Ensure compatibility with the existing system by enabling the import of student records exported from Mekelle University's current Ruby on Rails platform.

This system aims to provide a transparent and reliable solution for verifying academic credentials, reducing the risk of document forgery, and enhancing the university's credibility.

## **1.6 Scope of the Research**

This research focuses on the design, development, and implementation of a blockchain-based verification system for graduate students at Mekelle University. The scope is confined to the integration of Ethereum smart contracts and GraphQL technologies to enhance the security, accessibility, and trustworthiness of academic records.

The system developed in this study allows:

- Secure storage of graduate student records on the Ethereum blockchain using smart contracts.
- Uploading of multiple student records via CSV files and manual entry using a web-based form.
- Real-time querying and retrieval of student data by student ID using GraphQL APIs.
- A user-friendly interface built with React for administrators to manage records and for external parties to verify graduate information.

The research is confined to the verification of graduate student data only. The blockchain implementation is tested using a local Ethereum environment (Ganache) for development and demonstration purposes. The focus remains on showcasing the feasibility and advantages of using decentralized technologies for academic credential verification within the context of Mekelle University.

### **1.7 Significance of the study**

This study demonstrates the practical application of blockchain technology in the context of academic record verification at Mekelle University. By integrating Ethereum smart contracts with GraphQL, the research provides a secure, transparent, and tamper-proof system for verifying the credentials of graduated students. The significance of the study lies in its ability to address key challenges related to data authenticity, record manipulation, and third-party verification.

The system benefits various stakeholders in the following ways:

- For the University: It enhances institutional credibility by ensuring that graduation records are verifiable and immutable. It also reduces the administrative overhead involved in responding to verification requests.
- For Employers and Verifiers: It offers a decentralized and real-time method to confirm the authenticity of academic credentials without the need to contact the university directly.
- For Graduates: It empowers students with a reliable and permanent digital record of their achievements that can be accessed and verified globally.

- For the Academic Community: The study contributes to the growing body of research on blockchain applications in education, showcasing how decentralized technologies can be integrated with existing systems such as Ruby on Rails to improve transparency and trust.

The successful implementation of the system highlights the potential of combining smart contracts and GraphQL to modernize academic recordkeeping and verification processes, making them more efficient, secure, and accessible.

### **1.8 Contribution of the study**

The integration of Ethereum smart contracts with GraphQL offers a robust solution for modern academic records management. This system ensures secure, decentralized, and transparent verification of graduate data, bridging the gap between institutional data and external verification demands.

### **1.9 Thesis structure**

Chapter Two presents a review of the literature on Smart Contracts, GraphQL, blockchain, centralized databases, decentralized databases, and other related topics relevant to this thesis work from the perspective of student verification systems. Chapter three presents the methodology used in this thesis work. The fourth Chapter deals with the experimental setup and analysis for the proposed system. The fifth Chapter addresses conclusions and future work.

## **Chapter 2 : Literature Review**

### **2.1 Introduction**

Modern universities face challenges in securely managing and verifying academic credentials, especially for graduates applying to institutions or employers abroad. Traditional systems, like the one used at Mekelle University, often rely on centralized, database-driven architectures (e.g., Ruby on Rails with MySQL), which are prone to manipulation, loss, and inefficiencies in verification. Emerging technologies such as **blockchain** and **GraphQL** offer a decentralized, secure, and flexible alternative.

This literature review explores the current state of research on blockchain-based verification systems, the use of smart contracts for record immutability, and GraphQL for efficient data querying.

### **2.2 Blockchain Technology in Academic Credential Verification**

Blockchain is a decentralized and tamper-resistant ledger, widely regarded as a breakthrough in secure data storage and validation. Its applications in academic credential verification have been actively researched.[12] Proposed the concept of blockchain for lifelong learning records, enabling secure, verifiable storage of credentials.

Blockcerts, an open standard for blockchain credentials, was developed by MIT and Learning Machine [13], demonstrating real-world feasibility in educational settings. Similarly, [14]highlighted how blockchain improves trust in academic documents by reducing fraud and third-party verification delays.

In the African context, where digital infrastructure and trust are ongoing challenges, deploying such systems can significantly improve transparency and reduce corruption in educational certification [15]

### **2.3 Smart Contracts for Secure Academic Records**

Smart contracts are self-executing programs that run on the blockchain, ensuring the enforcement of pre-defined rules without human intervention. In education, smart contracts can enforce verification processes by allowing only authenticated actions such as issuing, viewing, and validating credentials.

Smart contracts eliminate the need for trusted intermediaries, making them ideal for academic records [16]. A decentralized framework using smart contracts for verifying academic transcripts, showcasing improved data integrity and autonomy [17].

The use of Ethereum smart contracts in academic systems has also been demonstrated in decentralized applications (DApps), where students, institutions, and verifiers interact transparently [18].

## **2.4 Decentralization Theory**

Decentralization is a key principle in distributed systems; tasks and decision-making are spread among multiple nodes, reducing reliance on any single point of control [19].

Ethereum smart contracts decentralize the verification process, removing the university as the sole authority, and storing credentials on a transparent, immutable ledger [20]. Instead, stakeholders like employers and embassies can independently query and validate records.

## **2.5 Data Integrity and Immutability**

Blockchain's fundamental promise is immutability. Once data is added, it cannot be altered. This principle ensures data integrity, which is critical in academic environments[21].

This theoretical basis supports the design of smart contracts that store student IDs, names, CGPAs, and departments in an immutable ledger, providing trust and transparency[22].

## **2.6 GraphQL for Data Access and Efficiency**

GraphQL, a query language developed by Facebook, allows clients to request only the data they need, making data fetching efficient, especially in applications interfacing with blockchain-based systems. Unlike REST, GraphQL allows nested queries and flexible schema definitions, enhancing developer productivity and system performance[11].

According to [23] GraphQL's expressiveness makes it suitable for structured data models often used in academic systems. Moreover, combining GraphQL with Ethereum smart contracts,

according to [24], allows for seamless integration of decentralized data with modern frontend applications.

This aligns well with the objectives of the proposed system for Mekelle University, where efficient querying of student credentials is crucial for user experience and performance.

While the integration of these technologies presents numerous advantages, challenges such as the need for robust infrastructure and user education must also be addressed to ensure successful implementation.

### **2.7 Existing Student Verification Systems**

Conventional student verification systems often rely on centralized databases, making them vulnerable to data tampering and requiring manual verification processes. Mekelle University's current system is built using Ruby on Rails with a MySQL database, which lacks features like decentralized validation, tamper-proof storage, and public verification access [25].

Some modern verification systems, like Digitary and TrueRec, offer digital credentials, but they depend on centralized services and do not utilize blockchain or smart contracts for validation. These systems provide limited transparency and control to students [26].

Blockchain-based systems, like those used by the University of Nicosia and Woolf University, offer decentralized degree verification, but they are proprietary and not easily adaptable for other institutions [9].

### **2.8 Web 3.0 in Blockchain-Based Systems**

Web 3.0, also known as Web3, represents the next generation of the internet that emphasizes decentralization, transparency, and user ownership of data through blockchain technology. Unlike Web 2.0, which depends on centralized servers and intermediaries, Web3 applications (DApps) operate on decentralized networks such as Ethereum, allowing users to interact directly with smart contracts without relying on trusted third parties [27].

This paradigm shift enhances trust, security, and immutability in data management and

transactions [28]. In educational systems, Web3 enables secure and verifiable digital credentials and academic record management, allowing graduates to independently prove their achievements without institutional mediation [29].

Thus, integrating smart contracts and GraphQL within a Web3 framework provides an innovative approach to ensuring secure, efficient, and decentralized access to academic data in higher education.

## **2.9 Blockchain Frameworks for Academic Verification Systems**

Blockchain frameworks provide the foundational architecture that supports decentralized applications by defining consensus algorithms, data models, and network protocols. Among the most prominent frameworks, Ethereum and Hyperledger have gained significant attention for their versatility in academic verification and data management systems. Ethereum is a public blockchain platform designed to execute smart contracts—self-executing scripts that automate agreement enforcement without intermediaries [22]. Its open and decentralized nature facilitates transparency and immutability, making it well-suited for student credential verification systems where trust and auditability are essential. In contrast, Hyperledger, developed under The Linux Foundation, operates as a permissioned blockchain framework, emphasizing privacy, scalability, and governance within organizational boundaries [30]. This makes it ideal for universities that require controlled data sharing while maintaining institutional autonomy. According to Tapscott and Tapscott [31], such frameworks extend Blockchain’s potential far beyond cryptocurrencies by supporting business logic, identity management, and academic record validation. Their flexibility enables integration with modern technologies like GraphQL and smart contracts, allowing seamless querying and interaction between blockchain data and university information systems. Consequently, both Ethereum and Hyperledger play a central role in ensuring transparency, efficiency, and trustworthiness in decentralized academic management systems.

## **2.10 Blockchain Security Mechanisms**

Blockchain ensures security through a combination of cryptographic, consensus, and decentralization mechanisms that collectively protect data integrity and authenticity. The primary layer of protection is cryptographic hashing, where each block references the previous one using algorithms such as SHA-256, ensuring immutability and tamper-resistance [21]. Any attempt to modify stored data would alter the hash values and be immediately detected across the network. In addition, digital signatures and asymmetric encryption safeguard user identity and transaction validity, ensuring that only authorized entities can add records [32]. These kinds of features make blockchain particularly suitable for sensitive applications such as educational record verification systems, where data confidentiality, integrity, and authenticity are crucial.

### **2.11 Cryptocurrency Wallets and Gas Fees**

In blockchain-based systems, a cryptocurrency wallet is a digital tool that enables users to securely store, send, and receive crypto-assets and interact with smart contracts. Each wallet contains a public key, used as an address to receive assets, and a private key, which serves as a digital signature for transaction authorization [33]. In Ethereum-based platforms, wallets such as MetaMask provide an interface for users to interact with decentralized applications (DApps), including those for academic record verification.

Every transaction executed on a blockchain requires a **gas fee**, which is a small amount of cryptocurrency paid to compensate miners or validators for processing and validating the transaction [27]. Gas fees are measured in **Gwei** (a subunit of Ether), and the total cost depends on the computational complexity of the smart contract and current network congestion[34].

### **2.12 Smart Contract Programming Languages**

Several programming languages have been developed to implement smart contracts on blockchain platforms. Among them, Solidity is the most widely used language for Ethereum-based applications.

Solidity is a high-level, contract-oriented programming language designed specifically for writing smart contracts that run on blockchain platforms such as Ethereum. It provides

developers with a syntax similar to JavaScript, enabling them to define contract structures, manage data types, and enforce transaction rules directly on the blockchain. Solidity allows for the creation of decentralized applications (DApps) by enabling automated execution of predefined conditions without intermediaries, thereby ensuring transparency, security, and immutability of records. Its support for inheritance, libraries, and complex user-defined types makes it a powerful tool for implementing logic in decentralized ecosystems such as student record systems, financial applications, and supply chain management [35].

### **2.13 Research Gap Analysis**

Existing studies on blockchain-based student verification systems largely focus on credential issuance or digital diplomas. However, limited research integrates Ethereum blockchain with GraphQL to enhance flexible data querying, especially in the context of Ethiopian higher education institutions. The identified gap is the absence of an efficient querying layer combined with immutable on-blockchain storage to support automated verification. This study aims to fill this research gap by proposing and evaluating an integrated Ethereum blockchain and GraphQL framework.

### **2.14 Research Gap**

Despite the significant potential of blockchain and smart contracts in academic credential verification, several limitations persist:

- **Lack of Integration Between Smart Contracts and Modern Querying Tools:** Integration with GraphQL is underexplored.
- **Limited Use in Developing Countries' Institutions:** Most blockchain-based student verification platforms are built in and for developed countries, with minimal adaptation for African universities like Mekelle University.
- **No Unified System Combining Blockchain, Smart Contracts, and GraphQL:** There is currently no open source or academic project that combines smart contracts for secure data management with GraphQL for efficient data querying and frontend integration, especially targeting university student verification systems.

This thesis addresses these gaps by designing and implementing a blockchain-based student verification system for Mekelle University that combines Ethereum smart contracts with GraphQL for efficient querying and modern web interface integration.

Several systems have attempted blockchain-based academic verification. Learning Machine and MIT's Blockcerts [13], and projects like EduCTX [24], have laid the foundational work. However, most lack real-time query capabilities and flexible integration for third-party services or UI/UX frameworks.

Additionally, these systems often focus on issuance and overlook ease of access for verifiers. Very few studies integrate GraphQL with smart contracts for education, highlighting a gap in efficient, decentralized data access solutions in academic settings, particularly in African universities.

The literature highlights the effectiveness of blockchain in securing academic credentials, while GraphQL enhances data querying and application performance. However, limited work exists on their integration, particularly in the African context and in student verification systems. This research addresses this gap by developing a GraphQL-integrated smart contract system for Mekelle University to verify graduated students securely and efficiently.

The table below shows the gaps that has grasped empirically.

| No | Author Name, Place, and Year  | Title   | Gap   |
|----|---|---|---|
| 1  | Manoj KUMAR D S(PhD)<br>2023  | Educational Certificate Verification System Using Blockchain  | The system is limited to certificate-level certification and does not manage complete student academic records. It lacks a flexible querying layer (GraphQL), does not support CSV/manual data import. No qualitative or performance evaluation is provided, and the study offers no implementation or testing within the Ethiopian higher education context. |
| 2  | Anushka Chaurasia (PhD)<br>International Journal of Computer Applications<br><b>July 2024</b> | Blockchain-based Authentication and Verification System for Academic Certificate using QR Code and Decentralized Applications | The system focuses only on certificate authentication, not full student record management. It relies on QR-code-based verification, with no flexible querying layer   |

|   |  |  |   |
|---|--|--|---|
|   |  |  | such as GraphQL, and does not support CSV upload from an existing system. No implementation or testing is done in the Ethiopian university context. |
| 3 | Henok Berhane Tadesse<br>Addis Ababa University<br><b>2021</b> | Block skill framework for a Mutable blockchain-based data: in case of a job-driven education system. | Focused only on the implementation of Blockchain. Doesn't consider which API to use for better performance.   |

*Table 2.1 List of related papers with their gap*

### **2.15 Theoretical Framework**

This research is grounded in a combination of Constructivist Learning Theory, Decentralization Theory, and Data Integrity Theory. When applied to graduate student verification systems, these theories collectively ensure the authenticity, transparency, and reliability of academic credentials. The theoretical framework thus provides a solid foundation for understanding how blockchain and GraphQL technologies can address long-standing challenges in academic credential verification and enhance trust in digital education systems.

Decentralization Theory further strengthens this foundation by promoting distributed control and trustless validation. Traditional student record systems, such as those built on centralized databases like the existing Ruby on Rails-based system at Mekelle University, are prone to data manipulation, system failure, and restricted accessibility. Blockchain technology mitigates these risks by decentralizing data storage and validation processes. Through smart

contracts, validation of grades and certificates becomes automated, transparent, and verifiable by all stakeholders without a central intermediary.

Finally, Data Integrity Theory ensures that information remains accurate, consistent, and tamper-proof throughout its lifecycle. Applying this theory to graduated student verification enhances the credibility of the data stored on the blockchain. Once academic credentials are recorded through a verified GraphQL interface and stored on a blockchain, they cannot be altered without detection. This immutable and verifiable data structure assures employers, institutions, and graduates that the credentials are authentic and trustworthy. Together, these theories provide a robust conceptual foundation for integrating smart contracts and GraphQL into a reliable, transparent, and secure e-student grade verification system for Mekelle University.

### **2.16 Conceptual framework**

The conceptual framework of this research illustrates the integration of **smart contracts** and GraphQL to create a transparent, tamper-proof, and decentralized system for verifying graduated students at Mekelle University. The model demonstrates how student data exported from the existing Ruby on Rails–based system is transformed and stored on the Ethereum blockchain through smart contract functions, ensuring immutability and public verifiability. GraphQL serves as a flexible query interface between the blockchain layer and the user interface, enabling real-time data retrieval and verification of graduated students' credentials.

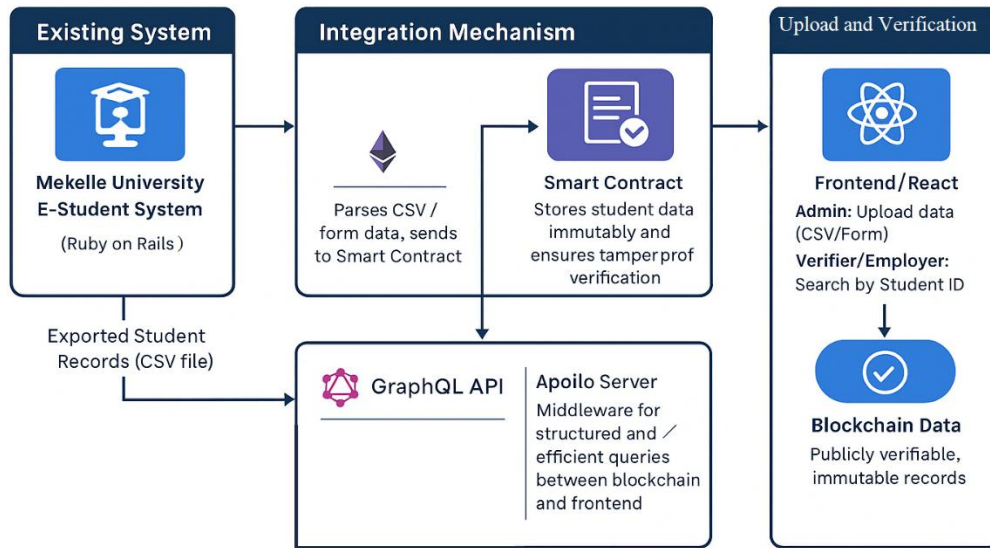


Figure 2.1 Conceptual framework of the system

## 2.17 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is a comprehensive legal framework established by the European Union to govern the collection, processing, and protection of personal data[36]. Although this research is conducted in Ethiopia, GDPR remains a global benchmark for data privacy standards and is frequently referenced in academic discussions related to digital systems, blockchain, and data verification. GDPR emphasizes principles such as transparency, data minimization, accuracy, storage limitation, integrity, and confidentiality when handling information personally [37]. In decentralized systems such as blockchain, GDPR raises unique challenges, particularly regarding the immutability of stored data and the right to be forgotten, since blockchain records cannot be easily altered or deleted [29]. For student verification systems, GDPR focuses on the importance of protecting sensitive academic records, enforcing strict access control, using hashed identifiers instead of raw personal data, and ensuring secure data transmission. Integrating GDPR principles strengthens the proposed blockchain and GraphQL solution by aligning it with internationally recognized privacy and data-protection standards, promoting trust, transparency, and compliance in handling student academic information.

The Ethiopian Computer Crime Proclamation No. 958/2016 outlines prohibitions against unlawful access, data interference, and misuse of personal information stored in digital systems [38], [39]. In addition, the Digital Strategy for Ethiopia 2025 highlights the importance of adopting global data-privacy standards and strengthening cybersecurity as the country transitions toward digital governance[39]. Recent national discussions on a draft Personal Data Protection Proclamation also indicate Ethiopia’s commitment to establishing GDPR-aligned legal protections for citizens’ personal data. For blockchain-based student information systems, these frameworks emphasize the need for secure data storage, confidentiality, controlled access, and proper authorization when managing sensitive academic records. Aligning the proposed smart contract and GraphQL architecture with both GDPR and Ethiopia’s evolving digital-governance policies reinforces the system’s acceptability and ensures that student data is handled responsibly and ethically.

## **Chapter 3 : Methodology**

This study employed a design and implementation-based methodology to develop a blockchain-powered student verification system by integrating Smart Contracts and GraphQL. The approach involved analyzing the existing student grade management system at Mekelle University, identifying limitations related to record verification, and designing a decentralized application (DApp) that enables secure storage and retrieval of graduated student data.

This research followed an inductive approach, as it aimed to explore and understand user experiences, perceptions, and practical challenges related to implementing a blockchain-based student verification system at Mekelle University. Rather than testing a pre-defined hypothesis, the study focused on collecting and analyzing qualitative data to generate themes and insights.

The methodology consisted of the following key stages:

### **3.1 System Design and Architecture**

- The system was architected as a decentralized application (DApp), consisting of three core components:
  - A smart contract deployed on a local Ethereum blockchain.
  - A GraphQL API for structured data querying.
  - A front-end interface for user interaction.
- The architecture supports importing graduated student data exported from the existing Ruby on Rails system and storing it immutably on the blockchain.

### **3.2 Smart Contract Development**

- A smart contract was written in Solidity and deployed using Hardhat and Ganache on a local Ethereum test network.
- The contract defines functions to add student records and retrieve them based on a unique student ID.

### **3.3 Smart Contract Platform Selection**

Selecting the appropriate blockchain platform is a critical methodological decision that directly influences the security, performance, and reliability of the proposed student verification system. Several platforms were considered, including Ethereum, Binance Smart Chain (BSC), Polygon, Solana, and Hyperledger Fabric. The selection process followed both qualitative and quantitative evaluation criteria to ensure that the chosen platform aligns with the requirements of decentralization, data integrity, interoperability, and long-term sustainability.

#### **3.3.1 Qualitative Comparison of Blockchain Platforms**

Ethereum was selected primarily due to its maturity, stability, and extensive ecosystem support. Unlike many emerging platforms, Ethereum provides a well-established environment for building smart contracts, supported by high decentralization and transparent governance. The platform benefits from a large global community and a rich set of development tools such as Hardhat, Truffle, and Ethers.js, which simplifies testing, debugging, and integration with technologies like Node.js, GraphQL, and React [27].

Compared to Binance Smart Chain and Polygon, which operate using fewer validators and more centralized control mechanisms, Ethereum provides stronger security guarantees and tamper resistance—critical features when handling sensitive academic records [27], [22]. Platforms like Solana offer higher transaction throughput but have experienced periodic outages and rely on more centralized validator structures, making them less suitable for applications that demand reliability and auditability [40]. Permissioned systems such as Hyperledger Fabric provide high throughput but lack the decentralized trust model required for external academic verification without institutional intermediaries.

#### **3.3.2 Quantitative Comparison and Empirical Justification**

Quantitatively, Ethereum demonstrates strong performance across key blockchain metrics. Its Proof-of-Stake (PoS) consensus mechanism is secured by thousands of globally distributed validators, providing high resistance to consensus manipulation and making attacks

economically unfeasible [41]. Ethereum finalizes transactions within approximately **12–15 seconds**, which is sufficient for academic verification workflows that do not require real-time execution.

Although some platforms such as Solana or Avalanche offer faster confirmation times (sub-second to a few seconds), these gains come at the cost of reduced decentralization and increased reliance on specialized hardware and centralized validator clusters [40]. Moreover, Ethereum’s economic security measured by staked value and market capitalization surpasses most competing public blockchains, reinforcing its suitability for systems requiring high assurance and long-term integrity [22].

### **3.3.3 Final Justification for Choosing Ethereum**

Based on both qualitative and quantitative evaluation criteria, Ethereum provides the most balanced solution in terms of security, decentralization, reliability, and developer support. These attributes make it suitable for a trusted academic verification system where transparency, immutability, and verifiability are essential. Furthermore, the availability of widely adopted frameworks and tools supported a smoother implementation process, ensuring that the proposed student verification system is both technically feasible and methodologically sound.

### **3.4 Data Integration and Upload**

- Student records were imported through:
  - A CSV file upload feature for bulk data addition.
  - A manual form for entering individual student records.
- The uploaded data was parsed using Node.js and sent to the smart contract via Ethers.js.

### **3.5 GraphQL API Implementation**

- A GraphQL API was built using Apollo Server to allow flexible and efficient querying of student records from the blockchain.
- Queries could be made to search for a student by ID and retrieve the corresponding graduation details.

### **3.6 Frontend Development**

- A user-friendly interface was developed using React.js, enabling:
  - CSV upload and form-based input of student data.
  - Real-time verification of student credentials by entering the student ID.

### **3.7 Testing and Validation**

- The sample system was tested in a local blockchain environment using Ganache.
- Functionality such as adding records, querying data, and verifying accuracy was validated through test cases and sample records exported from the Ruby on Rails-based system.

This methodology ensured a functional and secure blockchain-based system capable of verifying graduated students' records, addressing integrity, authenticity, and accessibility in academic credential verification

### **3.8 Research Design Justification**

A mixed-method approach combining qualitative analysis and experimental evaluation was adopted. The qualitative component provides insights into the challenges of the existing verification process, while the experimental component evaluates the performance and reliability of the proposed system. This design was selected because it enables both exploratory understanding and empirical testing of the solution.

### **3.9 Constructivism in Educational Technology**

Constructivist learning theory suggests that knowledge is actively constructed by learners based on their interactions with information. Applied to system design, this supports the need for verifiable, self-controlled educational records and empowers graduates with ownership over their academic data[42]

### **3.10 Institutional Theory**

Institutional theory highlights how organizations adopt innovations to gain legitimacy[43]. As blockchain adoption grows, Mekelle University's adoption of such systems aligns with institutional development.

This adoption reflects a strategic move to align with global trends and expectations in higher education transparency and innovation.

### **3.11 Interview Data Analysis Method**

The interview responses were coded and analyzed using thematic analysis [44]. Each response was reviewed to extract key concepts and patterns, which were then grouped into five themes. Frequencies and representative quotes were used to support the analysis part of this thesis work.

### **3.12 System analysis and design**

This chapter will present the detailed analysis and design of a blockchain-based graduated student verification system at Mekelle University. The system integrates Smart Contracts and GraphQL to ensure the secure and tamper-proof verification of student data, addressing existing limitations in the current Ruby on Rails-based system.

### **3.13 System Analysis**

#### **3.13.1 Existing System Overview**

The current student grade management system at Mekelle University is developed using Ruby on Rails. It manages student academic records and stores data in a centralized database. However, the system lacks a secure and decentralized method for third-party verification of graduate students. The verification process is manual and using a centralized database system, time-consuming, and vulnerable to data manipulation or forgery.

#### **3.13.2 Identified Problems**

- Lack of transparency and trust in the verification process.

- Centralized data storage creates a single point of failure.
- The manual verification process is inefficient and slow.
- Difficulty in validating records for external institutions or employers.

| <b>Criteria</b>              | <b>Ruby on Rails System<br/>(Currently Used)</b>                       | <b>Blockchain-Based System<br/>(Our Solution)</b>                       |
|------------------------------|--|---|
| System Architecture          | Centralized server-client model  | Decentralized blockchain architecture                                   |
| Data Storage                 | Stored in the MySQL, SQLite or other databases.                        | Stored on the Ethereum blockchain                                       |
| Security & Tamper Resistance | Susceptible to server-level tampering or data loss                     | Immutability ensured by smart contracts                                 |
| Transparency                 | Limited to internal users  | Transparent and verifiable to external users (employers, institutions)  |
| Cost of Operation            | Lower for internal use, but higher for long-term auditing/verification | Cost of transactions (gas fees), but higher trust and less human effort |

*Table 3.1 Comparison between Ruby on Rails System and Blockchain-Based Student Verification System*

Based on the identified limitations, the system was designed to meet the following functional and non-functional requirements:

### **3.13.3 Functional Requirements:**

- Import student data from a CSV file exported from the existing system through a web form.
- Adding student data using the input form
- Store graduated student records on the blockchain using a smart contract.
- Retrieve and verify student records by student ID.
- Provide a front-end interface for record input and verification.

### **3.13.4 Nonfunctional requirements:**

- Data immutability and security.
- Decentralized storage and trustless verification.
- Fast querying and response time.
- User-friendly web interface.

## **3.14 System Design**

### **3.14.1 System Architecture**

The system is designed as a Decentralized Application (DApp) with the following components:

- Frontend: Built with React.js for user interaction.
- Backend/API Layer: Apollo Server with GraphQL for querying blockchain data.
- Smart Contract Layer: Solidity smart contract deployed on a local Ethereum blockchain (Ganache).
- Blockchain Interaction: Ethers.js is used for communication between the frontend/backend and the Ethereum network.
- Data Source: Student records exported from the Ruby on Rails system in CSV format.

### **3.14.2 System Components and Workflow**

1. CSV Upload/Form Input:
  - Admins can upload student records via a CSV file or enter them manually through a form.
  - Data is parsed and validated using Node.js.
2. Smart Contract Interaction:
  - Parsed data is sent to a Solidity smart contract.
  - Records are stored immutably on the blockchain.
3. GraphQL API Layer:

- A GraphQL schema is defined to allow querying student records by student ID.
  - Apollo Server fetches data from the blockchain via Ethers.js.
4. Frontend Verification Interface:
- Users (e.g., employers or institutions) can search for a student using their ID.
  - Verification results are displayed instantly from blockchain data.

### 3.14.3 Data Flow Diagram

Below is an illustration of how data flows through our implemented system.

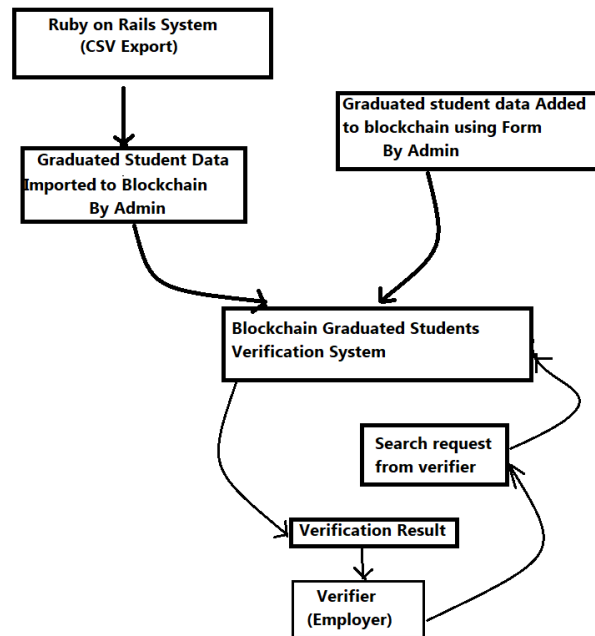


Figure 3.1 Dataflow diagram

### 3.14.4 Smart Contract Design

- The smart contract includes:
  - A struct to define student data (ID, Name, Department, CGPA, etc.).
  - Functions to add new students and retrieve records by ID.
- Data is stored in a blockchain database for immutability.

### 3.14.5 Security and Integrity Considerations

- Data stored on the blockchain can't be changed, ensuring record integrity.

- Read operations are public and do not require gas, enabling efficient verification.

This design ensures that student graduation records are verifiable, tamper-proof, and accessible for legitimate verification requests. By integrating Smart Contracts and GraphQL, the system achieves a high level of trust, automation, and performance compared to the traditional Ruby on Rails-based system.

### **3.14.6 Design Advantages**

- Decentralization: Smart contracts remove single points of failure [16].
- Efficiency: GraphQL minimizes data over fetching compared to other APIs.
- Transparency: Immutable blockchain records enhance trust among stakeholders.

### **3.14.7 Design Challenges**

- Cost: Mekelle University will need cryptocurrency and a cryptocurrency wallet for Blockchain transaction fees.
- Complexity: Integrating smart contracts and GraphQL introduces a steeper learning curve for IT staff.
- Latency: Blockchain interactions may slow down real-time operations compared to Rails. Because blockchains can take minutes to confirm transactions.

## Chapter 4 : Experimental setup and analysis

The implementation phase focused on building a decentralized application (DApp) that integrates Ethereum smart contracts and GraphQL APIs to verify and manage records of graduated students at Mekelle University. The goal was to replace centralized databases and manual verification processes with a secure, immutable, and transparent blockchain-based system. This section outlines the key components, tools, and workflows involved in the system's development.

### 4.1 Technology Used

- Smart Contract: Developed in Solidity and deployed on a local Ethereum network using Ganache.

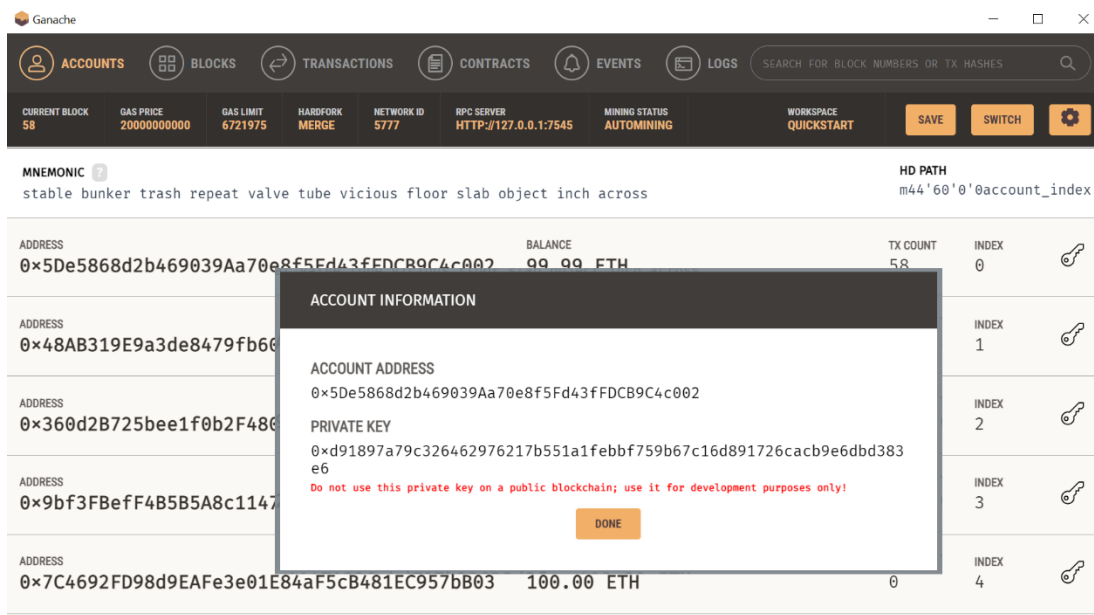


Figure 4.1: Local Ganache ETH Network

- Blockchain Interaction: Managed using Ethers.js within a Node.js backend.
- GraphQL API: Built using Apollo Server to expose blockchain data and operations.
- Frontend: Developed using React.js, providing an intuitive interface for administrators to add data and users to verify graduated students.

- CSV Integration: Used for bulk importing student records exported from the legacy Ruby on Rails system at Mekelle University.
- Form Input: Allows manual entry of individual student records into the blockchain.

## 4.2 Architecture Used

The architecture consists of three main layers:

1. Smart Contract Layer: Handles storage and access control for student records. Each record includes fields such as student ID, name, department, and CGPA..
2. API Layer: GraphQL serves as the middleware, querying the smart contract and returning structured data to the frontend.
3. Presentation Layer: The React interface includes:
  - A CSV upload tool to import multiple student records (Option 1).
  - A form interface to add individual students (Option 2).
  - A search function by student ID to retrieve and display records.

The screenshot displays a web interface with three main sections:

- Option One:** Titled "Uploading Graduated Students to the Blockchain", it indicates data is imported from a CSV file from an existing Ruby on Rails system. It features a blue "Upload CSV File" button with a sub-section containing a "Choose File" button (showing "No file chosen") and an "Upload" button.
- Option Two:** Titled "Registering Graduated Students to the Blockchain", it prompts the user to use a form below. The form includes four input fields: "Student ID:" (ID), "Student Full Name:" (Full Name), "Field/Department:" (Department), and "Comulative GPA:" (Comulative GPA). An "Add Student" button is positioned below these fields.
- Students Verification from Blochchain:** A green bar at the bottom containing an "Enter Student ID" input field and a "Search" button.

Figure 4.2: Home page interface using React.

### 4.3 Data Flow

1. Importing Student Data: CSV files exported from the Ruby on Rails system are parsed in Node.js and passed to the smart contract in batches.
2. Manual Entry: Admins can add students via a form that interacts with the smart contract (When necessary and approved by management).
3. Querying Records: A student ID is submitted through the frontend, and GraphQL queries the blockchain to retrieve and display verified data.

### 4.4 Results

The implemented system was successfully tested using student records from Mekelle University. Below are the major achievements and observed outcomes:

#### 4.4.1 System Functionality

- CSV Upload: Multiple graduate student records were imported and stored on the blockchain without data loss.

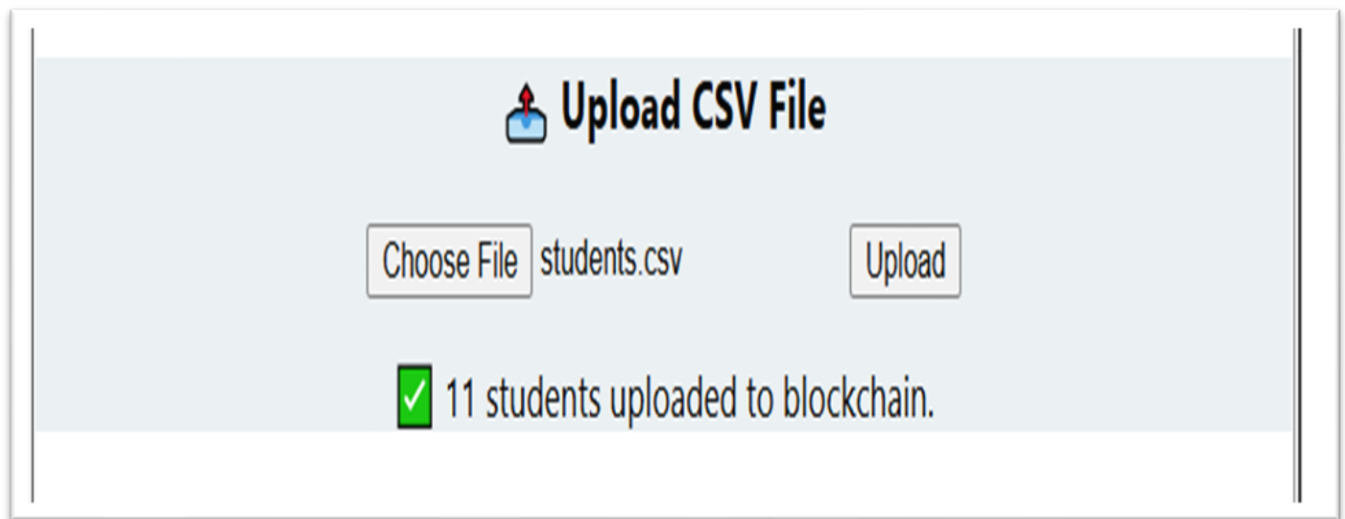


Figure 4.3: Successful Uploading CSV File.

- Form-Based Entry: Allowed manual addition of records when CSV import was not practical.

**Option Two**

🎓 Registering Graduated Students to the Blockchain 🎓

Using the form below

|  |                         |
|--|-------------------------|
| <b>Student ID:</b>                         | mu/pr112233/24          |
| <b>Student Full Name:</b>                  | Yonas Hadush Teway      |
| <b>Field/Department:</b>                   | MSc Network Engineering |
| <b>Comulative GPA:</b>                     | 3.99                    |
| <input type="button" value="Add Student"/> |                         |


localhost:3000 says  
Student added!

Figure 4.4: Successfully adding Student data

- Search and Verification: Student records were accurately retrieved using student ID, with data integrity ensured by the immutability of the blockchain.

🎓 **Students Verification from Blockchain** 🎓

mu/pr1211/24     



**Mekelle University**  
መቐለ ዩኒቨርሲቲ

**Verified Student Certificate**  
የተረጋገጠ የተማሪ ሰርቲፊኬት

|   |
|---|
| <b>We hereby certify that Mr./Ms.</b> Kiflom Solomon Berhe                                |
| <b>Student ID:</b> mu/pr1211/24   |
| <b>From Department/Field :</b> MSc. Suply Chain Management                                |
| <b>With Comulative GPA :</b> 3.9  |
| <b>is a graduate of our university.</b>   |
| <b>This record has been verified from our secure blockchain database and is accurate.</b> |

Figure 4.5: Successful searching (Verifying) a student from the blockchain using ID

- GraphQL Integration: Enabled precise and efficient data querying, simplifying the frontend interaction with complex blockchain operations.
- Performance: The system demonstrated fast transaction processing in the local development environment (Ganache). Real-world deployment on a testnet or mainnet would require optimization, particularly regarding gas fees and latency.

#### **4.4.2 Comparison to the Existing System**

The current student grade management system at Mekelle University, built using Ruby on Rails, lacks features for secure verification of graduates outside the institution. The blockchain-based system was introduced:

- Decentralized Verification: Eliminated the need for third-party validation or institutional intermediaries.
- Tamper-Proof Records: Once added, student data cannot be altered or deleted, enhancing credibility.
- Transparency: External parties such as employers can verify student records without contacting the university.

#### **4.4.3 User Experience**

- The user interface was intuitive and responsive.
- Verification was reduced to a few seconds, contrasting with the manual process that could take days (When an official is sent to the requesting institute using P.O. Box).

#### **4.4.4 Limitations**

- The current implementation runs on a local blockchain (Ganache); production deployment would require a secure testnet or mainnet environment.
- Gas fees and smart contract optimization need to be addressed for large-scale use.

#### **4.4.5 System Limitations and Data Integrity Considerations**

One of the critical limitations of blockchain-based systems is the immutability of stored records. While immutability ensures integrity and prevents unauthorized modification, it also means that incorrect information cannot be edited or deleted once written to the blockchain. If wrong student data is inserted, such as an incorrect ID, name, or graduation status, the system cannot simply update the original record. Instead, correction must be handled through append-only mechanisms, where a new transaction is added to replace the incorrect entry while keeping the original record visible for auditability. This approach preserves transparency while ensuring that users rely on the most recent, validated data state. Researchers emphasize that blockchain systems must therefore incorporate strong validation procedures, multi-layer verification, and access-control mechanisms to minimize the risk of inaccurate data being committed to the ledger[45]. In the context of a student verification system, this underscores the importance of proper data screening and institutional approval workflows before any record is permanently stored on the blockchain database.

#### **4.5 Comparative Performance Evaluation**

To assess the effectiveness of the proposed system, performance comparisons were made between:

1. Traditional manual verification,
2. The centralized Ruby on Rails system, and
3. The proposed Ethereum–GraphQL solution.

Evaluation metrics included query response time, transparency, data modification risk, and operational efficiency. Results show that while manual and centralized systems suffer from processing delays and vulnerability to data alteration, the blockchain-based system offers instant verification and immutable record storage. GraphQL significantly reduces over-fetching and under-fetching compared to REST-based data retrieval[46]. This means GraphQL eliminates the common over-fetching problem found in traditional REST APIs. While REST queries extract all data from a specific resource, GraphQL allows clients to request exactly the data they need in a single query.

## 4.6 Interview Data Analysis Approach

The interview data were analyzed using thematic analysis, following the six-step process by Braun and Clarke: familiarization, coding, generating themes, reviewing themes, defining themes, and reporting. Codes were grouped and refined into themes reflecting shared perceptions across stakeholder groups[44] .

### 4.6.1 Response Statistics and Analysis

| Stakeholder Group         | Respondents |
|---------------------------|-------------|
| Students                  | 5           |
| Academic & Registry Staff | 5           |
| External Stakeholders     | 5           |
| University IT Staff       | 5           |
| <b>Total</b>              | <b>20</b>   |

Table 4.1: Stakeholders participated in the interview

### 4.6.2 Summary Insights (with Statistics)

| Theme                | Summary   | Groups                        | Support |
|----------------------|---|-------------------------------|---------|
| Ease of Use          | Most users found the system simple, especially the CSV upload and verification  | Admin, Students, Stakeholders | 17/20   |
| Security & Trust     | Blockchain data is trusted as tamper-proof and authentic  | All groups                    | 20/20   |
| Fast Verification    | Record verification is now instant; huge time saver   | Admin, Stakeholders, Students | 16/20   |
| Technical Challenges | Concerns over staff training, cost, and rollback mechanisms   | IT Staff, Admin               | 12/20   |
| Adoption Support     | While most respondents supported university-wide and external adoption of the blockchain-based system, one stakeholder (R022) expressed conditional support. They emphasized the need for government oversight and assurance before relying | All groups                    | 19/20   |

| Theme | Summary  | Groups | Support |
|-------|--|--------|---------|
|       | <p>on the technology:</p> <p><i>“I do not know smart contracts or blockchain. The government must first make sure it’s immutable and issue formal instructions for us to trust and use online verification.”</i></p> <p>This feedback highlights the importance of policy, education, and national-level directives in achieving full trust and adoption, especially among non-technical stakeholders.</p> |        |         |

Table 4.2: Summary Insights (with Statistics)

### 4.6.3 Results and Discussion

Twenty participants from four groups (IT staff, administrative users, external stakeholders, and students) were interviewed to evaluate the new blockchain-based verification system. Thematic analysis revealed five dominant themes:

- **Ease of Use:** Participants appreciated the intuitive interface and automation features. For example, one admin said, “CSV upload eliminated paperwork and saved hours every week.”
- **Security & Trust:** Nearly all users cited confidence in the tamper-proof nature of blockchain. A student mentioned, “This ensures competitive fairness by preventing the use of falsified certificates.”
- **Fast Verification:** Especially among admins and employers, users emphasized that “verification now takes seconds instead of days.”
- **Technical Challenges:** IT respondents flagged concerns such as gas fees, skill shortages, and immutability issues. One IT member said, “We need skilled blockchain staff for deployment.”

- Adoption Support: The majority supported institutional adoption, except for 1 of the 20 respondents, with a stakeholder suggesting, “Other universities should use this system.”

| Theme                | Mentioned by        | Sample Quote   |
|----------------------|---------------------|--|
| Ease of Use          | Admin, Students, IT | "Simple and fast interface; no training needed."                 |
| Security & Trust     | All Groups          | "Immutable data ensures record authenticity."                    |
| Fast Verification    | Admin, Stakeholders | "No delay, it verifies instantly."                               |
| Technical Challenges | IT Staff, Admin     | "Need training."   |
| Support for Adoption | All Groups          | "Should be used university-wide and even by other institutions." |

Table 4.3 Thematic Summary

#### 4.6.4 Key Statistics

The interviews revealed strong support for the system’s usability, security, and future adoption. The most widely supported theme was trust in blockchain, with 100% of respondents agreeing it makes records tamper-proof.

95% of participants said the university should adopt the system, though one stakeholder (R022) emphasized the need for government-led assurances.

#### 4.6.5 Summary

A total of 20 respondents participated in the interview process, including IT staff (5), academic/registry staff (5), external stakeholders (5), and students (5). Their responses were thematically analyzed and converted into measurable insights. The following summarizes the key statistical findings across major evaluation themes:

- Ease of Use: 85% of respondents reported that the system was easy to use, especially highlighting the CSV upload and student search features.

- Trust in Blockchain: 100% of participants expressed trust in the Blockchain's ability to secure and protect academic data against tampering or falsification.
- Verification Speed: 80% agreed that the blockchain-based system provided faster verification compared to traditional methods.
- Support for Adoption: 95% of respondents supported full institutional adoption of the system. One external stakeholder (R022) provided conditional support, citing the need for government regulation and assurance regarding blockchain security.
- Technical Readiness Concerns: 50% of IT and administrative staff raised concerns about technical readiness, especially regarding deployment, staff training, and smart contract management.

These findings indicate broad institutional support for the blockchain-based system, while also highlighting specific areas for improvement, particularly in the areas of technical capacity building and stakeholder education.

## **Chapter 5 : Conclusion and future work**

### **5.1 Discussion**

The implementation of a blockchain-based student verification system represents a significant advancement in managing and authenticating academic records. The integration of smart contracts and GraphQL into the existing student information workflow has shown clear benefits in transparency, data integrity, and automation.

The developed system effectively bridges the gap between the traditional Ruby on Rails-based student management system used at Mekelle University and a decentralized infrastructure. By allowing student data exported from the existing system to be imported into a blockchain environment, the system ensures that records of graduated students become tamper-proof, publicly verifiable, and persistently accessible.

The inclusion of both CSV upload functionality and a manual form input mechanism provides flexibility, catering to both bulk and individual data entry scenarios. This is particularly beneficial during the transition from the legacy system to the new blockchain-based solution.

Furthermore, the use of GraphQL proved to be a robust and efficient solution for interacting with the smart contract. Unlike RESTful APIs, GraphQL allows precise queries, reducing the amount of data transferred and simplifying frontend development. This was particularly useful for the search functionality, where users needed to retrieve student information based on the student ID.

The use of Ganache for local blockchain testing, along with Ethers.js and Apollo Server, enables rapid development and debugging [9]. However, the local environment also highlights areas that require further consideration for real-world deployment, including network latency, gas fees, and scalability on public Ethereum networks.

### **5.2 Conclusion**

This thesis has successfully demonstrated the feasibility and effectiveness of using blockchain technology, specifically Ethereum smart contracts, in conjunction with GraphQL, to enhance

the management and verification of graduated student records supporting Mekelle University's digital transformation. The author recommends further deployment studies on public blockchain networks and integration with institutional digital ID frameworks.

Key achievements include:

- Development of a smart contract for storing and managing graduate data.
- Integration of data from the existing Ruby on Rails system via CSV import.
- A user-friendly React interface that allows searching, adding, and uploading student records.
- Use of GraphQL for structured, efficient querying of blockchain data.

By introducing decentralization into academic verification, this system reduces the risk of data manipulation, simplifies access to trusted records, and strengthens the credibility of the university's certification processes.

This work not only modernizes student record verification but also serves as a prototype for wider applications of blockchain in higher education. Future work can include integration with national or international credentialing systems, improved access controls using cryptographic methods, and deployment on scalable blockchain platforms.

### **5.3 Research Contributions**

This study contributes to academia and practice in several ways:

- **Theoretical Contribution:** Extends blockchain literature by demonstrating how distributed trust and data integrity theories apply to academic record verification.
- **Methodological Contribution:** Introduces a mixed-method evaluation mode

### **5.4 Future Work**

- Deploy the system on the Ethereum testnet and later on the mainnet.
- Expand smart contracts to support grades per course and semester.
- Implement mobile app access for verification.

- Add role-based access control for the home page to upload CSV and to add via the form.

### **5.5 Recommendation**

The adoption of Web3 technologies is recommended for Mekelle University's digital transformation roadmap. Implementing Web3-based solutions can enhance interoperability, reduce verification time, and support decentralized student data ecosystems that align with emerging global education technology trends.

## Reference

- [1] T. Gkrimpizi, V. Peristeras, and I. Magnisalis, “*Classification of Barriers to Digital Transformation in Higher Education Institutions: Systematic Literature Review*,” 2023. doi: 10.3390/educsci13070746.
- [2] A. Chaudhuri and J. S. Foster, “*Symbolic security analysis of Ruby-on-Rails web applications*,” in *Proceedings of the ACM Conference on Computer and Communications Security*, 2010. doi: 10.1145/1866307.1866373.
- [3] J. Chapman, “*How Safe Is Your Data? Cyber-Security In Higher Education*,” *HEPI Policy note*, no. April, 2019.
- [4] S. A. Jawaid, “*Cyber Security Threats to Educational Institutes: A Growing Concern for the New Era of Cybersecurity*,” *International Journal of Data Science and Big Data Analytics*, vol. 2, no. 2, 2023, doi: 10.51483/ijdsbda.2.2.2022.11-17.
- [5] M. Seki, “*DRuby and Rinda: Implementation and application of distributed Ruby and its parallel coordination mechanism*,” *Int J Parallel Program*, vol. 37, no. 1, 2009, doi: 10.1007/s10766-008-0086-1.
- [6] H. Sun, X. Wang, and X. Wang, “*Application of blockchain technology in online education*,” *International Journal of Emerging Technologies in Learning*, vol. 13, no. 10, 2018, doi: 10.3991/ijet.v13i10.9455.
- [7] S. Chandra and A. Farisi, “*Comparative Analysis of RESTful, GraphQL, and gRPC APIs: Performance Insight from Load and Stress Testing*,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 14, no. 1, pp. 81–85, Jan. 2025, doi: 10.32736/sisfokom.v14i1.2315.
- [8] S. Ben Othman and M. Getahun, “*Leveraging blockchain and IoMT for secure and interoperable electronic health records*,” *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-95531-8.
- [9] Robert, “*How Is Blockchain Used to Keep Student Records*,” <https://disruptionhub.com/how-is-blockchain-used-to-keep-student-records/>.
- [10] Z. Zheng *et al.*, “*An overview on smart contracts: Challenges, advances and platforms*,” *Future Generation Computer Systems*, vol. 105, 2020, doi: 10.1016/j.future.2019.12.019.
- [11] V. Spasev, I. Dimitrovski, and I. Kitanovski, “*An Overview of GraphQL : Core Features and Architecture*,” *ICT Innovations Conference 2020*, no. 24-Sep-2020, 2020.
- [12] M. Sharples and J. Domingue, “*The blockchain and kudos: A distributed system for educational record, reputation and reward*,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. doi: 10.1007/978-3-319-45153-4\_48.
- [13] P. Schmidt, “*Blockcerts — An Open Infrastructure for Academic Credentials on the Blockchain*,” Mit Media Lab.
- [14] A. Alammary, S. Alhazmi, M. Almasri, and S. Gillani, “*Blockchain-based applications in education: A systematic review*,” 2019. doi: 10.3390/app9122400.
- [15] N. G. Kofi, A. Emmanuel, and A. Frank, “*Adoption of blockchain technology in the banking sector of Ghana: Opportunities and challenges*,” *African Journal of Business Management*, vol. 17, no. 2, 2023, doi: 10.5897/ajbm2022.9428.

- [16] K. Christidis and M. Devetsikiotis, “*Blockchains and Smart Contracts for the Internet of Things*,” 2016. doi: 10.1109/ACCESS.2016.2566339.
- [17] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, “*Blockchain Technology Use Cases in Healthcare*,” in *Advances in Computers*, vol. 111, 2018. doi: 10.1016/bs.adcom.2018.03.006.
- [18] E. Leka and B. Selimi, “*Development and evaluation of blockchain based secure application for verification and validation of academic certificates*,” *Annals of Emerging Technologies in Computing*, vol. 5, no. 2, 2021, doi: 10.33166/AETiC.2021.02.003.
- [19] G. Wang, “*SoK: Understanding BFT Consensus in the Age of Blockchains*,” *IACR Cryptology ePrint Archive*, no. 911, 2021.
- [20] I. Grishchenko, M. Maffei, and C. Schneidewind, “*A semantic framework for the security analysis of ethereum smart contracts*,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-319-89722-6\_10.
- [21] S. Squarepants, “*Bitcoin: A Peer-to-Peer Electronic Cash System*,” *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.3977007.
- [22] V. Buterin, “*A next-generation smart contract and decentralized application platform*,” *Etherum*, no. January, 2014.
- [23] Q. Stokkink and J. Pouwelse, “*Deployment of a Blockchain-Based Self-Sovereign Identity*,” in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, Jul. 2018, pp. 1336–1342. doi: 10.1109/Cybermatics\_2018.2018.00230.
- [24] M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, “*EduCTX: A blockchain-based higher education credit platform*,” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2789929.
- [25] A. Notodikromo, *Learn rails 6: Accelerated web development with ruby on rails*. 2020. doi: 10.1007/978-1-4842-6026-5.
- [26] A. Tariq, H. Binte Haq, and S. T. Ali, “*Cerberus: A Blockchain-Based Accreditation and Degree Verification System*,” *IEEE Trans Comput Soc Syst*, vol. 10, no. 4, 2023, doi: 10.1109/TCSS.2022.3188453.
- [27] G. Wood, “*Ethereum: a secure decentralised generalised transaction ledger*,” *Ethereum Project Yellow Paper*, 2014.
- [28] W. Deng, T. Huang, and H. Wang, “*A Review of the Key Technology in a Blockchain Building Decentralized Trust Platform*,” 2023. doi: 10.3390/math11010101.
- [29] F. Casino, T. K. Dasaklis, and C. Patsakis, “*A systematic literature review of blockchain-based applications: Current status, classification and open issues*,” 2019. doi: 10.1016/j.tele.2018.11.006.
- [30] HYPERLEDGER, “*Hyperledger Architecture, Volume 1 Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus*,” 2019.
- [31] D. Tapscott and A. Tapscott, “*The Blockchain Revolution & Higher Education*,” *Education Review*, vol. 52, no. 2, 2017.
- [32] E. Portmann, “*Rezension „Blockchain: Blueprint for a New Economy“*,” *HMD Praxis der Wirtschaftsinformatik*, vol. 55, no. 6, 2018, doi: 10.1365/s40702-018-00468-4.

- [33] W. Bolt, “Review of: *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction* TEXTO COMPLETO,” *J Econ Lit*, vol. 55, no. 2, 2017.
- [34] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, 2017. doi: 10.1109/BigDataCongress.2017.85.
- [35] G. Zheng, L. Gao, L. Huang, and J. Guan, *Ethereum Smart Contract Development in Solidity*. 2020. doi: 10.1007/978-981-15-6218-1.
- [36] J. Zhang, F. Hassandoust, and J. E. Williams, “Online Customer Trust in the Context of the General Data Protection Regulation (GDPR),” *Pacific Asia Journal of the Association for Information Systems*, vol. 12, no. 1, 2020, doi: 10.17705/1pais.12104.
- [37] European Union, “Regulation 2016/679 of the European parliament and the Council of the European Union,” *Official Journal of the European Communities*, 2016.
- [38] K. M. Yilma, “Cybercrime lawmaking and human rights in ethiopia,” *Mizan Law Review*, vol. 15, no. 1, 2021, doi: 10.4314/mlr.v15i1.3.
- [39] FDRE, “Digital Ethiopia 2025 – A Strategy for Ethiopia Inclusive Prosperity,” 2019.
- [40] M. Dabbagh, K. K. R. Choo, A. Beheshti, M. Tahir, and N. S. Safa, “A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities,” *Comput Secur*, vol. 100, 2021, doi: 10.1016/j.cose.2020.102078.
- [41] S. M. Skh Saad and R. Z. Raja Mohd Radzi, “Comparative Review of the Blockchain Consensus Algorithm Between Proof of Stake (POS) and Delegated Proof of Stake (DPOS),” *International Journal of Innovative Computing*, vol. 10, no. 2, 2020, doi: 10.11113/ijic.v10n2.272.
- [42] R. Kumar Shah, “Effective Constructivist Teaching Learning in the Classroom,” *Shanlax International Journal of Education*, vol. 7, no. 4, pp. 1–8, 2019, doi: 10.34293/education.v7i4.600.
- [43] W. R. SCOTT, “W. Richard SCOTT (1995), *Institutions and Organizations. Ideas, Interests and Identities.*,” *M@n@gement*, vol. 17, no. 2, 2014, doi: 10.3917/mana.172.0136.
- [44] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qual Res Psychol*, vol. 3, no. 2, 2006, doi: 10.1191/1478088706qp063oa.
- [45] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, “Blockchain Technology - BEYOND BITCOIN,” *Berkley Engineering*, 2016.
- [46] “Comparative Analysis of Rest and GraphQL Technology on Nodejs-Based Api Development,” 2021. doi: 10.11594/nstp.2021.0908.

## Appendixes

### Appendix A: Interview Questions



MEKELLE UNIVERSITY

ETHIOPIAN INSTITUTE OF TECHNOLOGY MEKELLE

School of Computing

Department of Software Engineering

Research Interview with Mekelle University staff, students, and other stakeholders

Interview Questions

#### A. For University IT Staff (Developers / System Administrators)

1. How familiar are you with blockchain technologies such as Ethereum and smart contracts?
2. What are your thoughts on the technical feasibility of integrating blockchain with the university's current Ruby on Rails-based system?
3. What potential challenges do you foresee in deploying the blockchain-based system on a live environment (e.g., mainnet or institutional network)?
4. How would you evaluate the GraphQL API compared to other API in terms of system integration and data querying?
5. What is your view on the maintainability and scalability of this new system?

#### B. For Academic Registry or Admin Users

1. How would you describe the ease of use of the CSV upload and manual entry features for adding graduate student records?
2. How does the verification process in the new system compare with the current manual or paper-based process?
3. Are you confident that the data stored on the blockchain is secure and tamper-proof?
4. What concerns do you have, if any, about relying on a decentralized system for student verification?
5. What additional features would you like to see implemented in future versions?

### C. For External Stakeholders (Employers, Verifiers, Accreditation Bodies)

1. How important is it to you to have independent and instant verification of student graduation records?
2. Was the process of searching and verifying a student record using the provided interface simple and clear?
3. Do you trust the information retrieved from the blockchain as authentic and reliable?
4. How would the availability of this system affect your trust in academic credentials issued by Mekelle University?
5. Would you recommend that other institutions adopt similar blockchain-based verification systems?

### D. For Students

1. How familiar are you with technologies like blockchain, Ethereum, or smart contracts?
2. How does the verification process in the new system compare with the current manual or paper-based process?
3. Have you tried verifying your own academic record or graduation status using the new system? If yes, how was the experience?
4. How would the availability of this system affect your future careers?
5. Do you feel that your academic records are secure and tamper-proof on the blockchain system?

### E. General Evaluation Questions for All Interviewees

1. What are the main benefits you observed from using the blockchain-based verification system?
2. What are the limitations or challenges you experienced during testing or usage?
3. Do you believe this system will improve transparency and reduce credential fraud?
4. How would you rate the user interface and experience (UI/UX) of the system?
5. Would you support the university's transition to a blockchain-based student record verification system?

## Appendix B: Smart Contract Security and Functional Audit Report

Project Title: **“Integrating Ethereum Blockchain and GraphQL for a Secure Graduate Verification System at Mekelle University”**

Smart Contract Language: Solidity

Blockchain Environment: Ganache (Ethereum Local)

Audit Tool: Manual Review + Hardhat Console Logs + Unit Tests

### B1. Contract Overview

- Contract Name: StudentRegistry
- Main Functions:
  - addStudent(string id, string name, string cgpa, string dep)
  - searchStudent (string id) returns (string, string, string)

### B2. Security Checks Performed

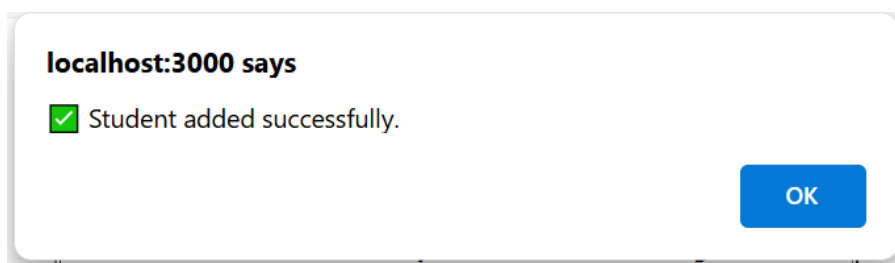
| Check                      | Status | Notes  |
|----------------------------|--------|--|
| Data Integrity             | Passed | Data written to the blockchain is immutable once added |
| Access Control             | N/A    | <b>Not implemented in this sample</b>                  |
| Integer Overflow/Underflow | N/A    | <b>Not implemented in this sample</b>                  |

### B3. Functional Tests Summary

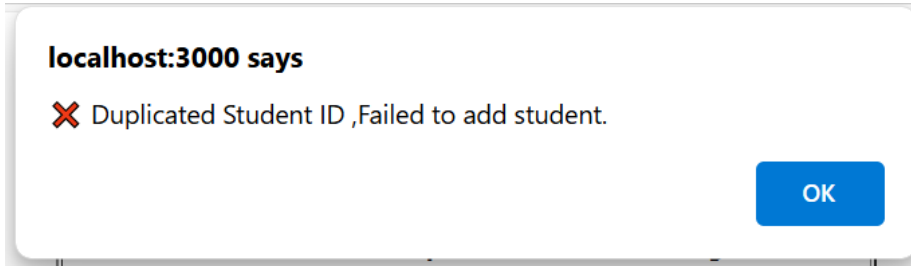
Tools Used: Hardhat + Mocha

Tests Executed:

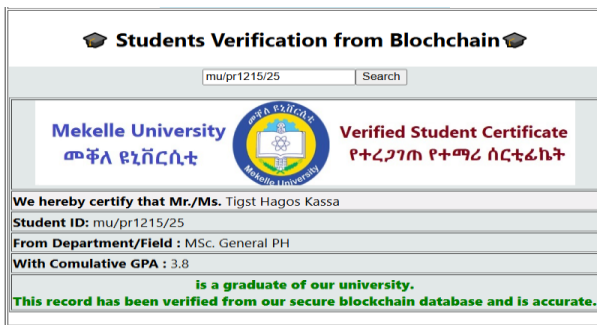
- Should add a student record successfully



- Should not allow duplicate student ID entries



- Should retrieve student data by ID



#### B4. Sample Transaction Hash (Ganache)

| CURRENT BLOCK   | GAS PRICE   | GAS LIMIT | HARDFORK | NETWORK ID                                 | RPC SERVER            | MINING STATUS | WORKSPACE  |
|---|-------------|-----------|----------|--|-----------------------|---------------|------------|
| 58  | 20000000000 | 6721975   | MERGE    | 5777                                       | HTTP://127.0.0.1:7545 | AUTOMINING    | QUICKSTART |
| TX HASH   |             |           |          |  |                       |               |            |
| 0x1d996eb491282bcc0cb79783000caab82d3964bd80f6d6e74594e8607660a196  |             |           |          |  |                       |               |            |
| FROM ADDRESS  |             |           |          | TO CONTRACT ADDRESS                        |                       | GAS USED      |            |
| 0x5De5868d2b469039Aa70e8f5Fd43fFDCB9C4c002                          |             |           |          | 0x3248C87E6b9DCBA0010B9d7f60FcDAb2F138Eede |                       | 95248         |            |
| <b>Student mu/pr1211/24 Added</b>                                   |             |           |          |  |                       |               |            |
| TX HASH   |             |           |          |  |                       |               |            |
| 0x3ab590304ecb833442fb3760e1d2ff8c09ac69ce02380b0c00aaf13d82e4cddf  |             |           |          |  |                       |               |            |
| FROM ADDRESS  |             |           |          | TO CONTRACT ADDRESS                        |                       | GAS USED      |            |
| 0x5De5868d2b469039Aa70e8f5Fd43fFDCB9C4c002                          |             |           |          | 0x3248C87E6b9DCBA0010B9d7f60FcDAb2F138Eede |                       | 35596         |            |
| <b>Student mu/pr1212/24 Added</b>                                   |             |           |          |  |                       |               |            |
| TX HASH   |             |           |          |  |                       |               |            |
| 0x9a48158deabe7c543843a7a5a2880920be28d867bfffa4f1bfe9baf630389acab |             |           |          |  |                       |               |            |
| FROM ADDRESS  |             |           |          | TO CONTRACT ADDRESS                        |                       | GAS USED      |            |
| 0x5De5868d2b469039Aa70e8f5Fd43fFDCB9C4c002                          |             |           |          | 0x3248C87E6b9DCBA0010B9d7f60FcDAb2F138Eede |                       | 35452         |            |
| <b>Student mu/pr1213/24 Added</b>                                   |             |           |          |  |                       |               |            |

The StudentRegistry smart contract was tested using the **Mocha framework** integrated in **Hardhat**. The tests were designed to verify core functionalities and enforce business rules. The following assertions passed.

```
StudentRegistry Contract
  ✓ should add a student (117ms)
  ✓ should not allow duplicate student ID (122ms)
  ✓ should revert if student ID not found

3 passing (3s)
```

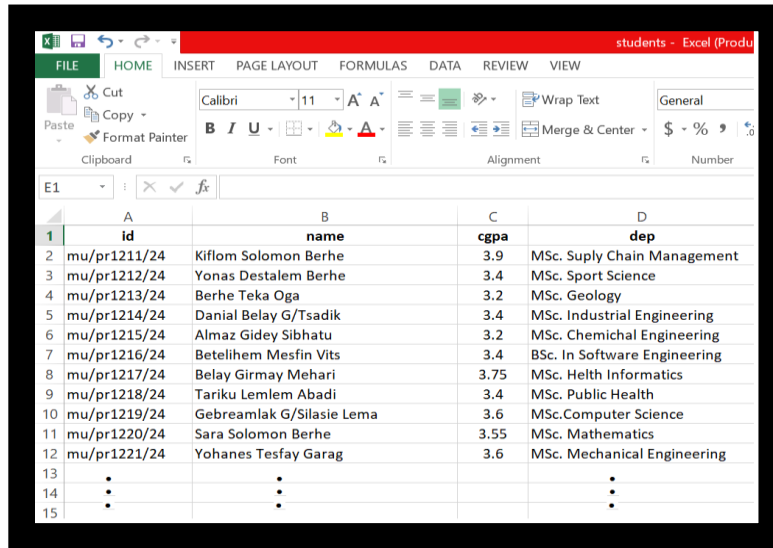
## B5. Conclusion

The smart contract was reviewed manually and tested using automated tools. The contract behaves as expected. All logic paths were validated using CSV and form inputs. The smart contract is safe to use in an academic record verification context on private or test networks.

## Appendix C: Manual of the Developed Sample System

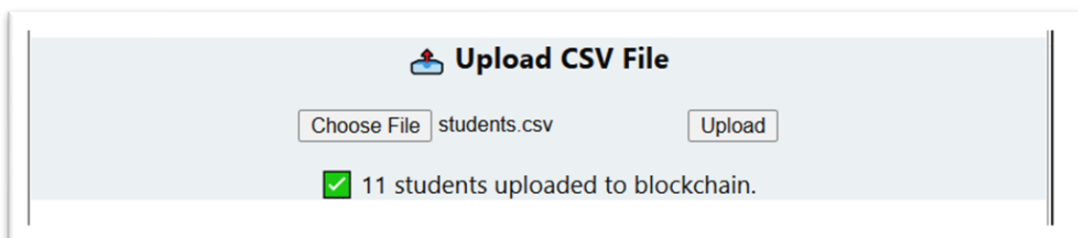
- Using Option 1

Prepare the CSV file from the existing system (Ruby on Rails)



|    | A            | B                         | C    | D                            |
|----|--------------|---------------------------|------|------------------------------|
| 1  | id           | name                      | cgpa | dep                          |
| 2  | mu/pr1211/24 | Kiflom Solomon Berhe      | 3.9  | MSc. Suply Chain Management  |
| 3  | mu/pr1212/24 | Yonas Destalem Berhe      | 3.4  | MSc. Sport Science           |
| 4  | mu/pr1213/24 | Berhe Teka Oga            | 3.2  | MSc. Geology                 |
| 5  | mu/pr1214/24 | Danial Belay G/Tsadiq     | 3.4  | MSc. Industrial Engineering  |
| 6  | mu/pr1215/24 | Almaz Gidey Sibhatu       | 3.2  | MSc. Chemichal Engineering   |
| 7  | mu/pr1216/24 | Betelihem Mesfin Vits     | 3.4  | BSc. In Software Engineering |
| 8  | mu/pr1217/24 | Belay Girmay Mehari       | 3.75 | MSc. Helth Informatics       |
| 9  | mu/pr1218/24 | Tariku Lemlem Abadi       | 3.4  | MSc. Public Health           |
| 10 | mu/pr1219/24 | Gebreamlak G/Silasie Lema | 3.6  | MSc.Computer Science         |
| 11 | mu/pr1220/24 | Sara Solomon Berhe        | 3.55 | MSc. Mathematics             |
| 12 | mu/pr1221/24 | Yohanes Tesfay Garag      | 3.6  | MSc. Mechanical Engineering  |
| 13 | .            | .                         | .    | .                            |
| 14 | .            | .                         | .    | .                            |
| 15 | .            | .                         | .    | .                            |

Then select the CSV file using Choose File and click Update.



11 Students are now successfully added to the blockchain  
Now we can search students by their ID (E.g., let us search the student ID with mu/pr1211/24 at the first row of the CSV file).

## 🎓 Students Verification from Blockchain 🎓

Mekelle University  
መቐለ ዩኒቨርሲቲ



Verified Student Certificate  
የተረጋገጠ የተማሪ ሰርቲፊኬት

We hereby certify that Mr./Ms. Kiflom Solomon Berhe

Student ID: mu/pr1211/24

From Department/Field : MSc. Suply Chain Management

With Comulative GPA : 3.9

**is a graduate of our university.**

**This record has been verified from our secure blockchain database and is accurate.**

- Using Option two

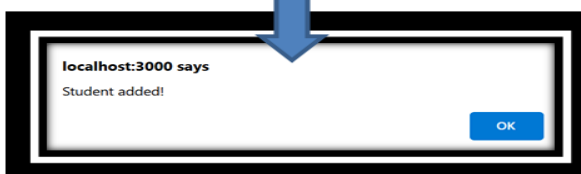
Using the second option, we can add students using the form.

### Option Two

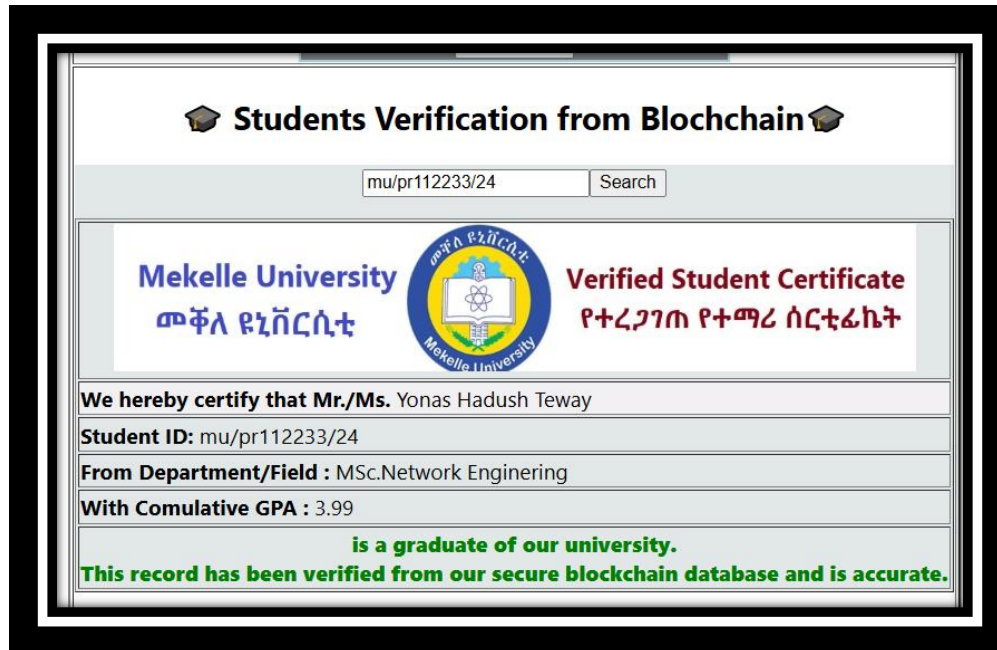
🎓 Registering Graduated Students to the Blockchain 🎓

Using the form below

|  |  |
|--|--|
| Student ID:                                | <input type="text" value="mu/pr112233/24"/>          |
| Student Full Name:                         | <input type="text" value="Yonas Hadush Teway"/>      |
| Field/Department:                          | <input type="text" value="MSc.Network Engineering"/> |
| Comulative GPA:                            | <input type="text" value="3.99"/>                    |
| <input type="button" value="Add Student"/> |  |



Now, we can search student ID mu/pr112233/24 from the blockchain.



#### Appendix D: Tools and Technologies Used

- Smart Contracts: Solidity
- Blockchain Network: Ganache (local Ethereum network)
- Development Frameworks: Hardhat, Node.js
- Blockchain Interaction: Ethers.js
- API Layer: GraphQL, Apollo Server
- Frontend: React.js
- Data Source Integration: CSV parser and manual entry form
- Existing System Data: Student records exported from the Ruby on Rails system

## Appendix E: Basic codes of the pilot project implemented

### The Smart contract written in solidity StudentRegistry.sol

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.20;

contract StudentRegistry {
    struct Student {
        string name;
        string cgpa;
        string dep;
    }

    mapping(string => Student) private students;
    mapping(string => bool) private idExists;

    function addStudent(
        string memory id,
        string memory name,
        string memory cgpa,
        string memory dep
    ) public {
        require(!idExists[id], "Student already exists");
        students[id] = Student(name, cgpa, dep);
        idExists[id] = true;
    }

    function getStudent(
        string memory id
    ) public view returns (string memory, string memory, string memory) {
        require(idExists[id], "Student not found");
        Student memory s = students[id];
        return (s.name, s.cgpa, s.dep);
    }
}
```

### The hardhat configuration file hardhat.config.js

```
• require("@nomicfoundation/hardhat-toolbox");
•
• module.exports = {
•   solidity: "0.8.20",
```

```

•   networks: {
•     ganache: {
•       url: "HTTP://127.0.0.1:7545",
•       accounts:
•         ["0x86b936210c457689af4977e67f9c400605f7f835db14e902557075df0c7542b8"]
•     }
•   }
• };

```

### The .env file

```

PRIVATE_KEY=0x86b936210c457689af4977e67f9c400605f7f835db14e902557075df0c7542b8
PROVIDER_URL='HTTP://127.0.0.1:7545'

```

### The resorver.js file

```

const { ethers } = require("ethers");
require("dotenv").config();

const StudentRegistry =
require('../backend/artifacts/contracts/StudentRegistry.sol/StudentRegistry.j
son');
const ABI = StudentRegistry.abi;
const CONTRACT_ADDRESS = "0x3D47063dE943A7De5E7d3adF52D341AF1e649af4";
const provider = new ethers.JsonRpcProvider("http://127.0.0.1:7545");
const wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider);
const contract = new ethers.Contract(CONTRACT_ADDRESS, ABI, wallet);

module.exports = {

Mutation: {
  addStudent: async (_, { id, name, cgpa, dep }) => {
    const tx = await contract.addStudent(id, name, cgpa, dep);
    await tx.wait();
    return true;
  }
},
Query: {
  getStudent: async (_, { id }) => {
    const [name, cgpa, dep] = await contract.getStudent(id);
    return { id, name, cgpa, dep };
  }
}
};

```

The deploy.js file that deploy the smart contract to contract address

```
const hre = require("hardhat");

async function main() {
  // Get the contract factory
  const StudentRegistry = await
hre.ethers.getContractFactory("StudentRegistry"); // Replace
"StudentRegistry" with your contract name

  // Deploy the contract
  const registry = await StudentRegistry.deploy(); // Add constructor
arguments if needed

  // Wait for the contract to be deployed
  await registry.waitForDeployment();

  // Get the deployed contract address
  const address = await registry.getAddress();
  console.log(`StudentRegistry deployed to: ${address}`);
  //console.log("☑ StudentRegistry deployed at:", contract.target);
  //console.log("☑ StudentRegistry deployed at:", contract.address);
}

main()
  .then(() => process.exit(0))
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

The uploadServer.js that upload csv file to the smart contract

```
const express = require("express");
const multer = require("multer");
const fs = require("fs");
const path = require("path");
const csv = require("csv-parser");
const { ethers } = require("ethers");
require("dotenv").config();

// ----- Smart contract setup -----
```

```

const StudentRegistry =
require('./backend/artifacts/contracts/StudentRegistry.sol/StudentRegistry.js
on');
const ABI = StudentRegistry.abi;
const CONTRACT_ADDRESS = "0x3D47063dE943A7De5E7d3adF52D341AF1e649af4";

const provider = new ethers.JsonRpcProvider("http://127.0.0.1:7545"); //
Ganache RPC
const wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider);
const contract = new ethers.Contract(CONTRACT_ADDRESS, ABI, wallet);

// ----- Express & Multer setup -----
const app = express();
const upload = multer({ dest: "uploads/" });

// Optional: Allow frontend requests (CORS)
const cors = require("cors");
app.use(cors({ origin: "http://localhost:3000" }));

// ----- CSV Upload Route -----
app.post("/upload", upload.single("file"), async (req, res) => {
  if (!req.file) return res.status(400).send("✘ No file uploaded.");

  const filePath = path.join(__dirname, "uploads", req.file.filename);
  const students = [];

  fs.createReadStream(filePath)
    .pipe(csv())
    .on("data", (row) => {
      // Skip empty or malformed rows
      if (!row.id || !row.name || !row.cgpa || !row.dep) return;

      students.push({
        id: String(row.id).trim(),
        name: String(row.name).trim(),
        cgpa: String(row.cgpa).trim(), // string required by contract
        dep: String(row.dep).trim(),
      });
    })
    .on("error", (err) => {
      console.error("CSV parse error:", err.message);
      return res.status(400).send(`✘ CSV parse failed: ${err.message}`);
    })
  });

```

```

.on("end", async () => {
  console.log(`Parsed ${students.length} students:`, students);

  const failed = [];

  for (const student of students) {
    try {
      const tx = await contract.addStudent(
        student.id,
        student.name,
        student.cgpa,
        student.dep
      );
      await tx.wait(); // Wait for transaction confirmation
      console.log("✅ Added:", student.id);
    } catch (err) {
      console.error(`❌ Failed to add ${student.id}:`, err.message);
      failed.push(student.id);
    }
  }

  if (failed.length > 0) {
    res.status(500).send(`⚠️ Uploaded but failed for IDs:
    ${failed.join(", ")}`);
  } else {
    res.send(`✅ ${students.length} students successfully uploaded`);
  }
});
});

// ----- Optional Test Route -----
app.get("/", (req, res) => res.send("🚀 Upload server running."));

// ----- Start Server -----
app.listen(5000, () => console.log("🚀 Upload server running on
http://localhost:5000"));

```

The check-student.js file that checks students by their ID in the blockchain database

```

const { ethers } = require("ethers");
require("dotenv").config();

```

```

const StudentRegistry = require('../student-
dapp/backend/artifacts/contracts/StudentRegistry.sol/StudentRegistry.json');
const ABI = StudentRegistry.abi;
const CONTRACT_ADDRESS = "0x3D47063dE943A7De5E7d3adF52D341AF1e649af4"; //

const provider = new ethers.JsonRpcProvider("http://127.0.0.1:7545");
const contract = new ethers.Contract(CONTRACT_ADDRESS, ABI, provider);

async function checkStudent(id) {
  try {
    const [name, age, course] = await contract.getStudent(id);
    console.log(`☑ Student Found:
ID: ${id}
Name: ${name}
Age: ${age}
Course: ${course}`);
  } catch (err) {
    console.error("✗ Error fetching student:", err.message);
  }
}

const studentId = process.argv[2]; // pass ID as command-line argument
if (!studentId) {
  console.error("! Provide a student ID: node check-student.js <id>");
  process.exit(1);
}
checkStudent(studentId);

```

The SearchStudent.jsx file that search students by their ID from the blockchain database

```

import './App.css';
import React, { useState } from "react";
import { gql, useLazyQuery } from "@apollo/client";

const GET_STUDENT = gql`
  query GetStudent($id: String!) {
    getStudent(id: $id) {
      id
      name
      cgpa
      dep
    }
  }
`

```

```

    }
  `;

export default function SearchStudent() {
  const [id, setId] = useState("");
  const [getStudent, { loading, data, error }] = useLazyQuery(GET_STUDENT);

  const handleSearch = () => {
    if (id.trim() !== "") {
      getStudent({ variables: { id } });
    }
  };

  return (
    <div class="div-2">
      <input
        placeholder="Enter Student ID"
        value={id}
        onChange={(e) => setId(e.target.value)}
      />
      <button onClick={handleSearch}>Search</button>

      {loading && <p>Loading...</p>}
      {error && <p style={{ color: "red" }}>Sorry, we can not find this ID
from MU blockchain database: {error.message}</p>}

      {data?.getStudent && (
        <div style={{ marginTop: "1rem" }}>
          <p>

            <table border="1">
              <tr>

                <td colspan="4" align="center"></img></td>

              </tr>

              <tr bgcolor="#f2ffr2f2">
                <td>
                  <strong>We hereby certify that Mr./Ms.</strong>
{data.getStudent.name}
                </td>

```

```

        </tr>
        <tr>
        <td>
        <strong>Student ID:</strong> {data.getStudent.id}
        </td>
        </tr>
        <tr>
        <td>
        <strong>From Department/Field :</strong> {data.getStudent.dep}
        </td>
        </tr>
        <tr>
        <td>
        <strong>With Cumulative GPA :</strong> {data.getStudent.cgpa}
        </td>
        </tr>

        <tr>
        <th align="center" >
        <strong> <font color="green">is a graduate of our
university.<br></br> This record has been verified from our secure blockchain
database and is accurate.</font></strong>
        </th>
        </tr>

    </table>
    </p>
</div>
    )}
</div>
);
}

```

The Addpage.js file that enables us to add and search graduate students(Front end page)

```

import React from "react";
import { ApolloClient, InMemoryCache, ApolloProvider } from "@apollo/client";
import AddStudent from "./AddStudent";
import SearchStudent from "./SearchStudent";
import UploadCSV from "./UploadCSV";

```

```

const client = new ApolloClient({
  uri: "http://127.0.0.1:4000",
  cache: new InMemoryCache()
});

function App() {
  return (
    <ApolloProvider client={client}>
      <div>
        <table border="2" align="center">
          <tr>
            <td align="center">
              <h1> Option One</h1>
              <h4>🎓 Uploading Graduated Students to the Blockchain🎓 </h4>
              <h4>From CSV file Imported from the Existing</h4>
              <h4>Ruby on Rails System</h4>
              <UploadCSV/>
              <h1> Option Two</h1>
              <h4>🎓 Registering Graduated Students to the Blockchain🎓 </h4>
              <h4>Using the form below</h4>
              <AddStudent />
            </td>
          </tr>
          <tr>
            <td align="center">
              <h2>🎓 Students Verification from Blockchain🎓 </h2>
              <SearchStudent />
            </td>
          </tr>
        </table>
      </div>
    </ApolloProvider>
  );
}

export default App;

```

## **Appendix F: Interview Data**

### **University IT Staff (Developers / System Administrators)(5)**

1. How familiar are you with blockchain technologies such as Ethereum and smart contracts?
2. What are your thoughts on the technical feasibility of integrating blockchain with the university's current Ruby on Rails-based system?
3. What potential challenges do you foresee in deploying the blockchain-based system on a live environment (e.g., mainnet or institutional network)?
4. How would you evaluate the GraphQL API compared to other API in terms of system integration and data querying?
5. What is your view on the maintainability and scalability of this new system?
6. What are the main benefits you observed from using the blockchain-based verification system?
7. What are the limitations or challenges you experienced during testing or usage?
8. Do you believe this system will improve transparency and reduce credential fraud?
9. How would you rate the user interface and experience (UI/UX) of the system?
10. Would you support the university's transition to a blockchain-based student record verification system?

### **B. For Admin and Registrar (5)**

1. How would you describe the ease of use of the CSV upload and manual entry features for adding graduate student records?
2. How does the verification process in the new system compare with the current manual or paper-based process?
3. Are you confident that the data stored on the blockchain is secure and tamper-proof?

4. What concerns do you have, if any, about relying on a decentralized system for student verification?

5. What additional features would you like to see implemented in future versions?

6. What are the main benefits you observed from using the blockchain-based verification system?

7. What are the limitations or challenges you experienced during testing or usage?

8. Do you believe this system will improve transparency and reduce credential fraud?

9. How would you rate the user interface and experience (UI/UX) of the system?

10. Would you support the university's transition to a blockchain-based student record verification system?

### **C. For External Stakeholders (Employers, Verifiers, Accreditation Bodies) (5)**

1. How important is it to you to have independent and instant verification of student graduation records?

2. Was the process of searching and verifying a student record using the provided interface simple and clear?

3. Do you trust the information retrieved from the blockchain as authentic and reliable?

4. How would the availability of this system affect your trust in academic credentials issued by Mekelle University?

5. Would you recommend that other institutions adopt similar blockchain-based verification systems?

6. What are the main benefits you observed from using the blockchain-based verification system?

7. What are the limitations or challenges you experienced during testing or usage?

8. Do you believe this system will improve transparency and reduce credential fraud?

9. How would you rate the user interface and experience (UI/UX) of the system?

10. Would you support the university's transition to a blockchain-based student record verification system?

#### **D. For Students (5)**

11. How familiar are you with technologies like blockchain, Ethereum, or smart contracts?
12. How does the verification process in the new system compare with the current manual or paper-based process?
13. Have you tried verifying the student using the new sample system? If yes, how was the experience?
14. How would the availability of this system affect your future careers?
15. Do you feel that your academic records are secure and tamper-proof on the blockchain system?
16. What are the main benefits you observed from using the blockchain-based verification system?
17. What are the limitations or challenges you experienced during testing or usage?
18. Do you believe this system will improve transparency and reduce credential fraud?
19. How would you rate the user interface and experience (UI/UX) of the system?
20. Would you support the university's transition to a blockchain-based student record verification system?

### Interview Data

| Respondent ID | Group    | Q1 Response                        | Q2 Response         | Q3 Response                                       | Q4 Response   | Q5 Response   | Q6 Response                           | Q7 Response  | Q8 Response                           | Q9 Response   | Q10 Response   |
|---------------|----------|------------------------------------|---------------------|---|---|---|---------------------------------------|--|---------------------------------------|---|--|
| R001          | IT Staff | Only info, but not practically     | feasible            | Gas cost  | GraphQL is good   | Maintainable if modularized well                      | Verification without central reliance | No challenge   | Reduces the risk of fake certificates | Clear, functional, but better if an alternative search by name is included. | I strongly support it. It's the right move for digital governance. |
| R002          | IT Staff | No experience with this            | Yes, it is feasible | Gas cost may be high                              | I never use it  | It is scalable but may be high-cost.                  | Fast verification                     | Technically skilled staff in smart contracts may be required | No fake certificates                  | Clear and simple  | I strongly support it. It's an important idea.                     |
| R003          | IT Staff | No experience with smart contracts | feasible            | Training on blockchain is mandatory for IT staff. | I have information that it is a fast and new technology | Maintainable if the IT staff is trained on blockchain | Excellent for transparency            | We may have a lack of skilled staff on the                   | Reduces the risk of fake certificates | Clear, functional, but better if an alternative search                      | Yes, Strongly.   |

| Respondent ID | Group               | Q1 Response                            | Q2 Response                               | Q3 Response                            | Q4 Response  | Q5 Response                      | Q6 Response                                     | Q7 Response  | Q8 Response                                | Q9 Response  | Q10 Response   |
|---------------|---------------------|--|---|--|--|----------------------------------|---|--|--|--|--|
|               |                     |  |   |  | developed by FB                                      |                                  |   | blockchain   |  | by name is included.                               |  |
| R004          | IT Staff            | Only info, but not practical           | feasible                                  | Gas cost                               | GraphQL is good                                      | Maintainable if modularized well | Verification without central reliance           | No challenge   | Reduces the risk of fake certificates      | Clear and easy                                     | I strongly support it. It's the right move for digital governance. |
| R005          | IT Staff            | Only info, but not practical           | feasible                                  | Gas cost                               | GraphQL is good                                      | Maintainable if modularized well | Verification without central reliance           | Maybe we can have a shortage of technical skills in smart contract management. | Reduces the risk of fake certificates      | Simple to use                                      | I strongly support it. It's the right move for digital governance. |
| R011          | Admin and Registrar | The CSV upload feature is very helpful | Eliminates paperwork and signatures; this | Yes, Blockchain stores data immutably. | What if we upload the wrong data? We cannot edit it. | Access control by department     | Instant student verification is a major win. We | Need a training  | Yes, this one is very difficult to falsify | Simple and Stable, even for staff with little tech | Yes, we just need proper documentation and policy support          |

| Respondent ID | Group               | Q1 Response                            | Q2 Response  | Q3 Response                           | Q4 Response                                      | Q5 Response                             | Q6 Response   | Q7 Response     | Q8 Response                                | Q9 Response  | Q10 Response  |
|---------------|---------------------|--|--|---------------------------------------|--|---|---|-----------------|--|--|---|
|               |                     |  | system takes only seconds to validate.   |                                       |  | nt would help                           | save hours every week   |                 |  | background   |   |
| R012          | Admin and Registrar | The CSV upload feature is very helpful | Eliminates paper work and signature, this system takes only seconds to verify. | Yes Blockchain stores data immutably. | What if we upload wrong data? We cannot edit it. | Access control by department would help | Instant student verification is major win. We save hours every week | Need a training | Yes, this one is very difficult to falsify | Simple and Stable even for staff with little tech background | No, we have to know the technology first in detail before deployment. But the idea is excellent |
| R013          | Admin and Registrar | Both options are mandatory             | Eliminates paper work and signature, this system                               | Yes Blockchain stores data immutably. | What if we upload wrong data? We cannot edit it. | Access control by department would help | Good idea   | Need a training | Yes, this one is very difficult to falsify | Easy to use  | Yes, We just need proper documentation and policy support                                       |

| Respondent ID | Group               | Q1 Response                            | Q2 Response  | Q3 Response                           | Q4 Response  | Q5 Response                             | Q6 Response   | Q7 Response     | Q8 Response                                | Q9 Response  | Q10 Response  |
|---------------|---------------------|--|--|---------------------------------------|--|---|---|-----------------|--|--|---|
|               |                     |  | takes only seconds to verify.  |                                       |  |   |   |                 |  |  |   |
| R014          | Admin and Registrar | The CSV upload feature is very helpful | eliminates paper work and signature, this system takes only seconds to verify. | Yes Blockchain stores data immutably. | What if we upload wrong data? We can not edit it . | Access control by department would help | Instant student verification is major win. We save hours every week | Need a training | Yes, this one is very difficult to falsify | Simple and Stable even for staff with little tech background | Yes, We just need proper documentation and policy support |
| R015          | Admin and Registrar | The CSV upload feature is very helpful | eliminates paper work and signature, this system takes only                    | Yes Blockchain stores data immutably. | What if we upload wrong data? We can not edit it . | Access control by department would help | Instant student verification is major win. We save hours every week | Need a training | Yes, this one is very difficult to falsify | Simple and Stable even for staff with little tech background | Yes, We just need proper documentation and policy support |

| Respondent ID | Group        | Q1 Response   | Q2 Response          | Q3 Response   | Q4 Response        | Q5 Response   | Q6 Response  | Q7 Response   | Q8 Response          | Q9 Response        | Q10 Response   |
|---------------|--------------|---|----------------------|---|--------------------|---|--|---|----------------------|--------------------|--|
|               |              |   | seconds to verify.   |   |                    |   |  |   |                      |                    |  |
| R021          | Stakeholders | It is very important we need fast ,independent verification | It is clear and fast | Yes, if it is from blockchain it hasn't been edited or forged | It increases trust | Absolutely, other institutions should adopt like this | No need to request official from the university. We can verify documents in seconds. | No challenge in our side,but the university should have well-trained IT staff for this new technology | No face certificates | Simple and usable. | Yes strongly support   |
| R022          | Stakeholders | It is very important we need fast ,independent              | It is clear and fast | Yes, if it is from blockchain it hasn't been edited or forged | It increases trust | Absolutely, other institutions should adopt like this | No need to request official from the university. We can verify document              | No challenge in our side,but the university should have well-   | No face certificates | Simple and usable. | I am not familiar with blockchain or smart contracts. The government must ensure these systems are secure and make it official that online verification is |

| Respondent ID | Group        | Q1 Response   | Q2 Response          | Q3 Response   | Q4 Response        | Q5 Response   | Q6 Response  | Q7 Response   | Q8 Response          | Q9 Response        | Q10 Response                       |
|---------------|--------------|---|----------------------|---|--------------------|---|--|---|----------------------|--------------------|------------------------------------|
|               |              | verification  |                      |   |                    |   | s in seconds.  | trained IT staff for this new technology  |                      |                    | mandatory before we can rely on it |
| R023          | Stakeholders | It is very important we need fast ,independent verification | It is clear and fast | Yes, if it is from blockchain it hasn't been edited or forged | It increases trust | Absolutely, other institutions should adopt like this | No need to request official from the university. We can verify documents in seconds. | No challenge in our side,but the university should have well-trained IT staff for this new technology | No face certificates | Simple and usable. | Yes strongly support               |
| R024          | Stakeholders | It is very important we need fast                           | It is clear and fast | Yes, if it is from blockchain it hasn't been                  | It increases trust | Absolutely, other institutions should adopt like this | No need to request official from the university. We can                              | No challenge in our side,but the university   | No fake certificates | Simple and usable. | Yes strongly support               |

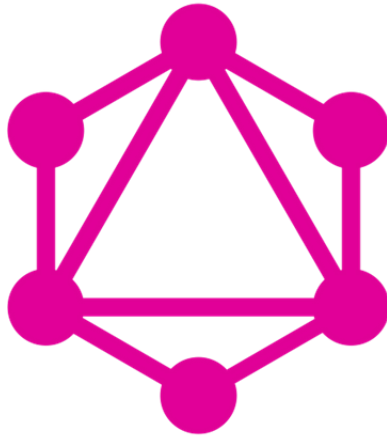
| Respondent ID | Group        | Q1 Response   | Q2 Response          | Q3 Response   | Q4 Response                | Q5 Response   | Q6 Response  | Q7 Response   | Q8 Response          | Q9 Response        | Q10 Response   |
|---------------|--------------|---|----------------------|---|----------------------------|---|--|---|----------------------|--------------------|--|
|               |              | ,independent verification                                   |                      | edited or forged  |                            |   | verify documents in seconds.   | should have well-trained IT staff for this new technology   |                      |                    |  |
| R025          | Stakeholders | It is very important we need fast ,independent verification | It is clear and fast | Yes, if it is from blockchain it hasn't been edited or forged | It increases trust         | Absolutely, other institutions should adopt like this | No need to request official from the university. We can verify documents in seconds. | No challenge in our side,but the university should have well-trained IT staff for this new technology | No fake certificates | Simple and usable. | Yes strongly support                                     |
| R031          | Students     | No information  | Good idea            | Very simple   | I can apply for a job with | Yes, but according to the                             | No fake certificates   | I am new for  | Exactly              | Excellent          | Yes, This ensures competitive fairness by preventing the |

| Respondent ID | Group    | Q1 Response                   | Q2 Response           | Q3 Response | Q4 Response                                     | Q5 Response   | Q6 Response   | Q7 Response | Q8 Response | Q9 Response | Q10 Response                  |
|---------------|----------|-------------------------------|-----------------------|-------------|---|---|---|-------------|-------------|-------------|-------------------------------|
|               |          |                               |                       |             | full confidence                                 | information you gave me, yes but I have no information about smart contract |   | blockchain  |             |             | use of falsified certificates |
| R032          | Students | A little about web3           | I appreciate the idea | Simple      | Increase confidence                             | Yes   | I need additional knowledge to judge  | I trust     | Yes         | Excellent   | Yes                           |
| R033          | Students | A little about cryptocurrency | Good idea             | Very easy   | Employers can verify our certificate in seconds | Yes   | I think it is good but ICT staff should be skilled on the technology used for better management | Yes         | Yes         | Very good   | Exactly                       |

| Respondent ID | Group    | Q1 Response            | Q2 Response                                  | Q3 Response   | Q4 Response                               | Q5 Response | Q6 Response                            | Q7 Response | Q8 Response | Q9 Response | Q10 Response |
|---------------|----------|------------------------|--|---|---|-------------|--|-------------|-------------|-------------|--------------|
| R034          | Students | Never heard about this | Appreciate the idea                          | No need high computer skill. Anyone can use it simply | No need more days for official transcript | Yes         | No challenge                           | Yes         | Yes         | Good        | Yes          |
| R035          | Students | No knowledge           | No need application for official transcript. | Easy to use   | We can apply with confidence              | Yes         | No challenge but mob app is mandatory. | Yes         | Yes         | Very good   | Yes          |



+



GraphQL