# NIRMALA MEMORIAL FOUNDATION COLLEGE OF SCIENCE AND COMMERCE

**KANDIVALI (E)-Mumbai**

# PROJECT

# ON

# Enhancing Academic Credentials Security And Verification Using Blockchain


**NAME: GOVIND RAMA PARAB**

**CLASS/DIVISION: M.Sc. IT SEMESTER: IV**

**ROLL NO.:_____UNIVERSITY SEAT NO: 1173226**

# Nirmala Memorial Foundation College of Commerce and Science

*Accredited by NAAC with B++ (First Cycle)*
*ISO 9001:2015 Certified*
*Recognized under the Section 2(F) & 12(B)*

# CERTIFICATE

This is to certify that **Mr. Govind Rama Parab** Seat no: **1173226**, student of **M.Sc. Information Technology - Semester IV** has successfully completed project titled **Enhancing Academic Credentials Security And Verification Using Blockchain** during the academic year 2023-24.

_____

Co-ordinator

_____ _____

External Examiner Seal and Date

**Table of Content**

## Introduction

In the realm of education and professional development, academic credentials are essential for verifying an individual's educational achievements and professional qualifications. However, the integrity of these credentials is often challenged by the limitations of traditional paper-based and digital formats. Paper-based credentials are vulnerable to forgery and manipulation, while digital credentials can be easily altered or deleted. This poses a significant challenge for employers, educational institutions, and other stakeholders who rely on these documents to make informed decisions. Blockchain technology, with its inherent properties of immutability, decentralization, and transparency, presents a compelling solution to address these shortcomings and enhance the security and verification of academic credentials.

### Background Study

Traditionally, the certification authority verifies diplomas or certificates because it is difficult to distinguish between genuine and fake certificates without specialized tools and knowledge that the certificate issuer can only provide [1]. Credential fraud in education, involving the falsification of credentials, poses challenges for institutions and employers. Common types include fake degrees, degree mills, transcript fraud, and credential misrepresentation. Fake degrees often use advanced printing techniques, requiring institutions to implement rigorous security measures. Degree mills grant degrees without academic requirements, necessitating awareness and background checks. Transcript fraud involves altering academic transcripts, mitigated by secure formats and blockchain technology. Credential misrepresentation extends to dishonesty in resumes and job applications, countered by thorough checks and promoting integrity. Vigilance is crucial to prevent consequences such as undermining academic integrity and eroding trust in educational institutions and the workforce. Cryptographic techniques used in Blockchain enhance the security and integrity of transactions recorded by a distributed ledger. Blockchain solves the problem of lack of trust by maintaining transaction records to each participating node. Transactions are recorded in a block which is added by a miner using a consensus algorithm. In addition, the Merkle tree generates a cryptographic fingerprint of the

entire set of transactions for a block to ensure its integrity and inclusion. The chain is created by storing the cryptographic fingerprint of the previous block [2]. Blockchain technology's potential in enhancing academic credential security gained momentum in 2015, addressing challenges in traditional verification methods. The Blockcerts project by MIT in 2015 pioneered tamper-proof digital academic credentials, marking a significant advancement. Subsequent platforms in 2016 aimed at transparent issuance, storage, and verification of academic records. Pilot projects in 2017 showcased blockchain's feasibility for managing and verifying academic records, leading to increased adoption. In 2018, the focus shifted to interoperability and standardization, fostering a cohesive credentialing ecosystem. Recognition expanded beyond education in 2019, drawing interest from governments and industry players. The COVID-19 pandemic in 2020 accelerated the adoption of blockchain for secure and efficient virtual credentialing. In 2021, developments prioritized user-centricity, data privacy, and integration with educational platforms. The concept expanded in 2022 to include various learning experiences, prompting research on blockchain's potential in admissions, student records, and lifelong learning pathways. Ongoing research aims to refine blockchain-based credentialing solutions, positioning blockchain to enhance the integrity, authenticity, and accessibility of academic credentialing.

**Aim**

The primary aim of this project is to develop and implement a blockchain-based solution that enhances the security, authenticity, and management of academic certificates, thereby eliminating forgery and tampering. This solution aims to provide a reliable and efficient verification process for educational institutions, employers, and other stakeholders, thereby enhancing trust and transparency in credential verification.

**Objective**

The primary objectives of this project are:

- To secure academic certificates against forgery and tampering through the conversion of certificates into digital signatures.

- To develop a blockchain-based system that ensures the immutability and verifiability of academic credentials.
- To create a transparent and decentralized verification process that can be easily accessed and trusted by all stakeholders.
- To integrate blockchain technology with existing educational and professional development platforms to streamline credential management.

**Scope**

The scope of the project encompasses the design, development, integration, and implementation of blockchain-based solutions tailored for enhancing the security and verification of academic credentials. This includes the establishment of secure data storage mechanisms, development of user-friendly interfaces, implementation, testing, documentation, and training. By addressing these components, the project aims to realize the potential of blockchain technology in revolutionizing the credentialing landscape and improving trust and transparency in academic credential verification processes.

**Applicability**

Using blockchain demonstrates significant applicability across various domains, addressing challenges related to fraud prevention, transparency, efficiency, data privacy, and stakeholder experience in credential verification processes. By harnessing the capabilities of blockchain technology, the project aims to modernize and secure credentialing systems while fostering trust and reliability in academic credentials. This technology can be particularly beneficial in the following areas:

- **Educational Institutions:** Universities and schools can issue tamper-proof digital diplomas and certificates.
- **Employers:** Companies can verify the academic credentials of job applicants with certainty.
- **Professional Certification Bodies:** Organizations can ensure the authenticity of professional qualifications and licenses.

- **Students and Professionals:** Individuals can confidently share their verified credentials with potential employers or institutions.

**Hardware and Software Requirements:**

**Hardware Requirements:**

- **Processor**: Core i5 (Minimum)
- **RAM**: 4GB (Minimum)

These hardware specifications ensure sufficient computing power and memory for running the development environment and executing blockchain-related tasks effectively.

**Software Requirements:**

- **Operating System**: Windows 10
- **Programming Language**: Python 3.9.2
- **Python 3.9.2**: Chosen for its versatility, readability, and extensive library support, essential for backend logic development and interaction with APIs.
- **Smart Contract Language**: Solidity
- **Solidity**: Essential for writing smart contracts on the Ethereum blockchain, defining rules and behaviors for decentralized applications (Dapps).

**Libraries**:

- **web3**: Enables interaction with Ethereum blockchain nodes, facilitating tasks such as data querying, transaction handling, and smart contract deployment.
- **requests**: Simplifies HTTP requests for communication with external APIs like Infura and Pinata, crucial for blockchain data retrieval and IPFS interactions.
- **ipfshttpclient**: Provides methods to interact with IPFS via HTTP, used in conjunction with the Pinata API for decentralized file storage associated with blockchain transactions.
- **firebase-admin**: Manages application data securely using Firebase services, supporting functionalities like user authentication, real-time database management, and cloud storage integration.
- **Platform Used**: Ethereum Sepolia Testnet

- **Ethereum Sepolia Testnet**: Utilized for testing and validating smart contracts and applications in a simulated Ethereum blockchain environment before deployment to the mainnet.

**Additional Tools and APIs:**

- **MetaMask**: Ethereum wallet provider used for managing accounts, transactions, and securely interacting with decentralized applications (Dapps).
- **Infura API**: Allows connection to the Ethereum blockchain without running a full node, providing scalable and reliable access.
- **Pinata API**: Used for interfacing with IPFS, ensuring secure storage and retrieval of decentralized file data associated with blockchain transactions.
- **Remix IDE**: Used for compiling the smart contract to get ABI and Bytecode of smart contract.

**Python** is a versatile programming language known for its simplicity and readability. It is widely used in various fields, including blockchain development. When combined with libraries like Web3, Python becomes a powerful tool for interacting with Ethereum blockchain networks.

**Blockchain Basics:**

Blockchain is a peer-to-peer system of transacting values with no trusted third parties in between.

- It is a shared, decentralized, and open ledger of transactions. This ledger database is replicated across a large number of nodes.
- This ledger database is an append-only database and cannot be changed or altered. It means that every entry is a permanent entry. Any new entry on it gets reflected on all copies of the databases hosted on different nodes.
- There is no need for trusted third parties to serve as intermediaries to verify, secure, and settle the transactions.
- It is another layer on top of the Internet and can coexist with other Internet technologies.

- Just the way TCP/IP was designed to achieve an open system, blockchain technology was designed to enable true decentralization. In an effort to do so, the creators of Bitcoin open-sourced it so it could inspire many decentralized applications.

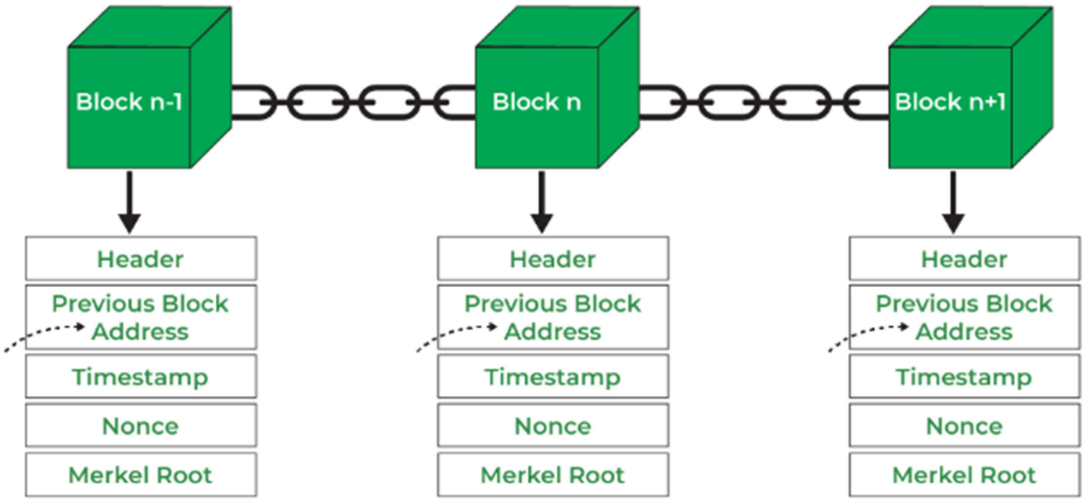
**Blockchain Data Structure:**

A blockchain is a type of digital ledger that keeps a continuously growing list of records, called blocks. These blocks are linked together and secured using cryptographic techniques. Each block in a blockchain contains several key elements:

- **Cryptographic Hash of the Previous Block:** This is a unique code that identifies the previous block in the chain. It links the blocks together in sequence, forming a continuous chain. The hash ensures that any change in one block would alter the hashes of all subsequent blocks, making tampering detectable.
- **Timestamp:** This records the exact time when the block was created, ensuring the chronological order of the blocks.
- **Transaction Data:** This includes the information or records that the block is intended to store, such as financial transactions, contracts, or other types of data.

**Structure of a Blockchain**

The blockchain is designed to ensure data security, transparency, and immutability. This is achieved through a data structure that consists of a sequence of blocks, where each block contains:



**Block Header:** The block header contains metadata about the block, including:

- **Block Height:** The position of the block in the blockchain, indicating how many blocks have come before it.
- **Timestamp:** The exact time when the block was created.
- **Previous Block Hash:** The cryptographic hash that links this block to the previous one, ensuring the continuity of the chain.

**Transaction Data:** The actual data being recorded on the blockchain, which can include anything from financial transactions to contracts or other types of records.

**Nonce:** A random number used in the Proof of Work (PoW) consensus algorithm. The nonce is critical for mining, as it ensures that creating a block is computationally difficult, but once created, the block is easy to verify. Miners adjust the nonce to generate a hash that meets the network's difficulty requirements.

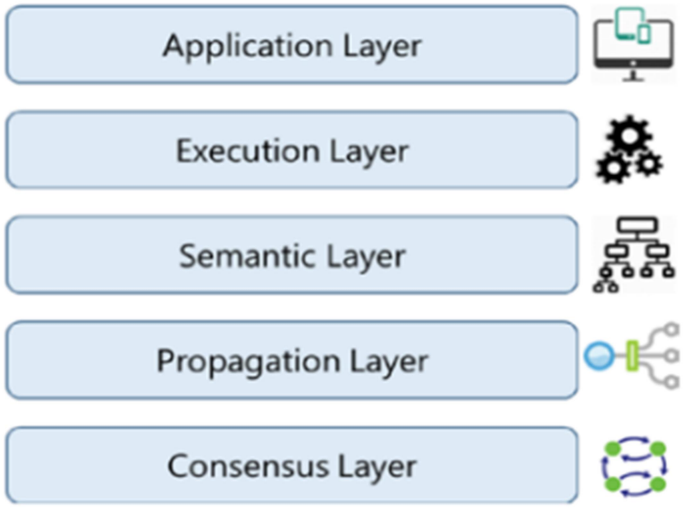
**Merkel Root:** It is a type of data structure frame of different blocks of data. A Merkel Tree stores all the transactions in a block by producing a digital fingerprint of the entire transaction. It allows the users to verify whether a transaction can be included in a block or not.

**Data Storage in Blocks:**

- **Transaction Grouping:** In a blockchain, multiple transactions are grouped together into a block. These transactions can range from financial operations to other types of data records.
- **Broadcasting and Verification:** When a new transaction occurs, it is broadcast to all participants (nodes) in the network. Each node then verifies the transaction for its validity.
- **Block Formation:** Once a sufficient number of transactions are verified, they are compiled into a new block. This block is then added to the blockchain, following the rules of the network’s consensus algorithm.
- **Decentralized Ledger:** Every participant in the network maintains a copy of the entire blockchain. This decentralized nature ensures that the data is resistant to tampering and censorship, as altering the blockchain would require the cooperation of the majority of the network's participants.

**Various Layers of Blockchain:**

The blockchain technology is based on a layered approach. There are few layers for the blockchain technology discussed in the following sections.




**1. Application Layer**

- The application layer is where a user can code the desired functionality and build the application for the user.
- Since blockchain is a decentralized technology with no server involved, the application needs to be installed on each node.
- Although there are some instances where blockchain is in the backend and the applications need to be hosted on a web server and require server-side programming, it would be preferable if there were no server involved in the blockchain network, as it would defeat the purpose and benefit of blockchain technology.

**2. Execution Layer**

- This layer handles the execution of all instructions performed at the application layer for all the nodes present on the blockchain network.
- The set of instructions could range from simple ones to multiple instructions. For example, a smart contract is a small code that needs to be executed when funds need to be transferred from one person to another.

- If an application is present on all the nodes of the blockchain network, the code has to be executed independently on all the nodes. To avoid inconsistencies in the output, the execution of code on a set of inputs should always produce the same output for all the nodes present on the blockchain.

**3. Semantic Layer**

- This layer is also called the logical layer of the blockchain layer suite. It deals with the validation of transactions performed in the blockchain network and also validates the blocks being generated in the network.
- When a transaction comes up from a node, the set of instructions are executed on the execution layer and gets validated on the semantic layer.
- The semantic layer is also responsible for the linking of the blocks created in the network.

**4. Propagation Layer**

- The propagation layer deals with peer-to-peer communications between the nodes, allowing them to discover each other and sync with other nodes in the network.
- When a transaction is carried out, it gets broadcast to all other nodes in the network. Also, when a node proposes a block, it will immediately get broadcast in the entire network so that other nodes can use this newly created block and work upon it.
- The propagation of the block or a transaction in the network is defined in this layer and ensures the stability of the complete network.
- However, depending upon the network capacity or network bandwidth, sometimes propagation could occur instantly, and other times it may take longer.

**5. Consensus Layer**

- This layer is the base layer for most blockchain systems. The main purpose of this layer is to ensure that all the nodes agree on a common state of the shared ledger.
- The layer also deals with the safety and security of the blockchain. There are many consensus algorithms that can be applied for the generation of cryptocurrencies like Bitcoin and Ethereum. They use a proof-of-work mechanism to select a node randomly out of the various nodes present on the network to propose a new block.

- Once a new block is created, it is propagated to all the other nodes to check if the new block is valid with the transactions in it. Based on the consensus from all other nodes, the new block gets added to the blockchain.

**Types of Blockchain**

**1. Private Blockchain Networks**

- Private blockchains operate on closed networks and tend to work well for private businesses and organizations.
- Companies can use private blockchains to customize their accessibility and authorization preferences, parameters to the network, and other important security options.
- Only one authority manages a private blockchain network.

**2. Public Blockchain Networks**

- Bitcoin and other cryptocurrencies originated from public blockchains, which also played a role in popularizing distributed ledger technology (DLT).
- Public blockchains help to eliminate certain challenges and issues, such as security flaws and centralization.
- With DLT, data is distributed across a peer-to-peer network, rather than being stored in a single location.
- A consensus algorithm is used for verifying information authenticity; proof of stake (PoS) and proof of work (PoW) are two frequently used consensus methods.

**3. Permissioned Blockchain Networks (Hybrid)**

- Also sometimes known as hybrid blockchains, permissioned blockchain networks are private blockchains that allow special access for authorized individuals.
- They combine the security and control of private blockchains with some transparency and flexibility, making them ideal for organizations needing structured access and privacy.
- Organizations typically set up these types of blockchains to get the best of both worlds, enabling better structure when assigning who can participate in the network and in what transactions.

**4. Consortium Blockchains**

- Similar to permissioned blockchains, consortium blockchains have both public and private components, except multiple organizations manage a single consortium blockchain network.
- Although these types of blockchains can initially be more complex to set up, once they are running, they can offer better security.
- Consortium blockchains are optimal for collaboration with multiple organizations.

**Advantages of Blockchain Technology:**

Blockchain is an emerging technology with many advantages in an increasingly digital world:

• **Highly Secure**

It uses a digital signature feature to conduct fraud-free transactions making it impossible to corrupt or change the data of an individual by the other users without a specific digital signature.

• **Decentralized System**

Conventionally, you need the approval of regulatory authorities like a government or bank for transactions; however, with Blockchain, transactions are done with the mutual consensus of users resulting in smoother, safer, and faster transactions.

• **Automation Capability**

It is programmable and can generate systematic actions, events, and payments automatically when the criteria of the trigger are met.

**Importance of Blockchain Technology**

- **Security:** Security is the primary concern for all kinds of online activities. Lots of data are stolen, and information is breached in this digital world. Blockchain provides a very high level of security, making it impossible to breach because of its decentralized nature.
- **Transparency:** Blockchain technology is very transparent as everything is visible to all participants from the beginning to date. One can see everything on the decentralized

network, making it a very open technology. It reduces the chance of any kind of discrepancy in the system because nothing is hidden.

- **Inexpensive:** Blockchain technology is the most reasonable financial model available right now. Compared to traditional economic models, it is much less expensive. Many companies are now looking to use blockchain technology because it can save significant costs in their economic models, especially in the banking industry.
- **Time of Transaction is Less:** Transactions using blockchain technology take very little time to complete, much faster than the time taken in traditional technology. Within a couple of minutes, one can send or receive financial documents and money without waiting for hours.
- **Increased Efficiency in Finance:** There is no involvement of any third party in blockchain technology, saving a lot of intermediary costs. All transactions happen directly from one individual to another. In the traditional banking system, the cost is higher to process financial transactions. Using blockchain technology, banks and companies can increase their economic efficiency.
- **Fraud Protection for Businesses:** Due to the high transparency of transactions in blockchain technology, any kind of fraud can be easily identified. Any fraud that occurs in the open-source ledger of Blockchain cannot stay hidden, ensuring businesses are always protected.
- **Increased Use of Blockchain Tokens:** Using Blockchain, a token can represent any piece of information. This includes an identity for an IoT device, instructions for an algorithm, origin information about a product, patents, a vote in an election, an energy kilowatt, a certificate credit, a digital ownership certificate, a share in a company, ownership of a house, and many more.
- **Scope of Innovation:** There is a massive scope in Blockchain technology because its features are open and programmable. It helps to rebuild systems in various fields, offering numerous possibilities for innovation. It can also reduce bureaucracy due to blockchain technology's transparency and efficiency.

- **No Middlemen in Transaction:** In blockchain technology, there is no need for mediators or intermediaries in transactions such as digital payments, insurance claims, asset management, the stock exchange, land registry, and more.
- **Numerous Applications of Blockchain:** Blockchain technology has many applications and uses in the future. Some possibilities in the field of Blockchain include digital currency, microfinance, P2P lending, remittance, global payments, e-commerce, smart contracts, escrow, wagers, etc. Other possibilities include digital rights, record-keeping, intellectual property, voting, ownership, title records, healthcare, securities, derivatives, crowd funding, debt handling, private markets, and equity markets.
- **Internet of Things:** Blockchain technology will significantly impact the Internet of Things (IoT). The identity of every device and the security of information for millions of connected devices has become crucial. Blockchain technology can manage data privacy, ownership protection, and the huge volume of data from devices. It can also serve as a base for developing new services, such as automatic supply chain services.
- **Smart Contracts:** A smart contract is a contract where specific situations and conditions are specified, helping execute a predefined task automatically. Blockchain technology is instrumental in automating predefined actions. The goal of smart contracts is to reduce transaction costs, enhance execution speed, and provide higher security compared to traditional law contracts. Applications of smart contracts include supply chain management, voting systems, healthcare data, personal information access, identity access, payment and rent agreements, royalty distribution agreements, and intellectual property rights.
- **High Flexibility of Usage:** Due to its high level of security and numerous applications, blockchain technology is very flexible in terms of usage. The cryptography used in Blockchain makes it extremely useful for executing transactions flexibly.
- **Decentralized Autonomous Organization (DAOs):** Blockchain technology allows for the development of Decentralized Autonomous Organizations (DAOs), which create value without human intervention. There is no need for a management team to make

decisions, and transactions are run by code. Smart contracts are also executed automatically when needed.

- **Sensitive Digital Records:** The healthcare industry is adopting blockchain technology to secure confidential digital records and gain full control over who can access how much data. Blockchain has become one of the top 5 priorities in the healthcare industry according to 40% of healthcare executives in a survey.
- **Huge Savings:** It is estimated that more than $100-$150 billion will be saved by 2025 due to the adoption of blockchain technology. This will help reduce costs in personnel, support functions, operations, IT, data breach-related incidents, and more. It also reduces the cost of counterfeit products and fraud.
- **Financial Privacy:** Many new crypto currencies are emerging, such as Beam, Monero, and Zcash, focusing on financial privacy. This means that information about economic ownership will be confidential and private. No individual will be able to access the details of another person's financial holdings.
- **Prevention of Data Leaks and Hacking:** Numerous incidents of hacking and data leaks have shaken people's trust in keeping their data and personal information with companies. However, with blockchain technology, data and information are highly secure, and there is no possibility of data leaks or hacking.

**Some Common Existing Use Cases of Blockchain**

- **Asset Registration and Transactions:** Any type of property or asset, whether physical or digital, such as laptops, mobile phones, diamonds, automobiles, real estate, e-registrations, digital files, etc., can be registered on blockchain. This enables these asset transactions from one person to another, maintains the transaction log, and checks validity or ownership. Also, notary services, proof of existence, tailored insurance schemes, and many more such use cases can be developed.
- **Financial Use Cases:** There are many financial use cases being developed on blockchain such as cross-border payments, share trading, loyalty and rewards systems, Know Your Customer (KYC) among banks, etc. Initial Coin Offering (ICO) is one of the most trending

use cases as of this writing. ICO is the best way of crowdsourcing today by using cryptocurrency as digital assets. A coin in an ICO can be thought of as a digital stock in an enterprise, which is very easy to buy and trade.

- **Collective Wisdom and Prediction Markets:** Blockchain can be used to enable "The Wisdom of Crowds" to take the lead and shape businesses, economies, and various other national phenomena by using collective wisdom! Financial and economic forecasts based on the wisdom of crowds, decentralized prediction markets, decentralized voting, as well as stocks trading, can be possible on blockchain.
- **Music Royalties:** The process of determining music royalties has always been convoluted. The internet-enabled music streaming services facilitated higher market penetration, but made the royalty determination more complex. This concern can pretty much be addressed by blockchain by maintaining a public ledger of music rights ownership information as well as authorized distribution of media content.
- **Internet of Things (IoT):** This is the IoT era, with billions of IoT devices everywhere and many more to join the pool. A whole bunch of different makes, models, and communication protocols makes it difficult to have a centralized system to control the devices and provide a common data exchange platform. This is also an area where blockchain can be used to build a decentralized peer-to-peer system for the IoT devices to communicate with each other. ADEPT (Autonomous Decentralized Peer-To-Peer Telemetry) is a joint initiative from IBM and Samsung that has developed a platform that uses elements of Bitcoin’s underlying design to build a distributed network of devices—a decentralized IoT. ADEPT uses three protocols: BitTorrent for file sharing, Ethereum for smart contracts, and TeleHash for peer-to-peer messaging in the platform.
- **Government Use Cases:** In the government sectors as well, blockchain has gained momentum. There are use cases where technical decentralization is necessary, but politically should be governed by governments: land registration, vehicle registration and management, e-Voting, etc. are some of the active use cases. Supply chains are another area where there are some great use cases of blockchain. Supply chains have

always been prone to disputes across the globe, as it was always difficult to maintain transparency in these systems.

**What is Web3?**

Web3 is a Python library specifically designed for interacting with the Ethereum blockchain. It provides an intuitive interface to connect to Ethereum nodes, deploy smart contracts, send transactions, and retrieve blockchain data.

**Key Concepts in Web3**

- **Blockchain Interaction**: Web3 allows developers to connect to Ethereum nodes using various protocols (HTTP, WebSocket, IPC) to interact with the Ethereum blockchain.
- **Smart Contracts**: It facilitates deploying, interacting with, and querying smart contracts deployed on the Ethereum blockchain.
- **Transactions**: Developers can create, sign, and send transactions using Web3, enabling interaction with Ethereum accounts and smart contracts.
- **Blockchain Data**: Web3 provides methods to retrieve blockchain data such as account balances, transaction histories, and smart contract states.
- **Event Handling**: It supports listening to and handling blockchain events, essential for building decentralized applications (Dapps) that react to changes on the blockchain.

**Why Use Python with Web3?**

Python's simplicity and extensive libraries make it a preferred language for blockchain development. Web3 abstracts the complexities of interacting with Ethereum, allowing developers to focus more on application logic rather than low-level blockchain protocols.

**What are smart contracts?**

Smart contracts are self-executing contracts with the terms of the agreement directly written into code, running on blockchain platforms like Ethereum. They automatically enforce and execute the terms of a contract when predefined conditions are met, removing the need for intermediaries and reducing the risk of fraud. Solidity is the most widely-used programming

language for writing smart contracts on Ethereum. Designed to target the Ethereum Virtual Machine (EVM), Solidity is a statically-typed, high-level language that resembles JavaScript in its syntax, making it accessible to many developers. It allows developers to create contracts for various applications, including decentralized finance (DeFi), supply chain management, and gaming, by providing the tools to define complex contract logic and interactions within the decentralized Ethereum network.

**What are Ethereum Testnets?**

Based on access control, blockchains can be classified as public and private. Public blockchains are also called permissionless blockchain and private blockchains are also called permissioned blockchains. The primary difference between the two is access control. Public or permissionless blockchains do not restrict addition of new nodes to the network and anyone can join the network. Private Blockchains have a limited number of nodes in the network and not everyone can join the network. Examples of public blockchains are Bitcoin and Ethereum main nets. An example of a private blockchain can be a network of a few Ethereum nodes connected to each other but not connected to the main net. These nodes would be collectively called a private blockchain. Private Blockchains are generally used by enterprises to exchange data among themselves and their partners and/or among their sub-organizations. When we develop applications for blockchains, the type of blockchain, public or private, makes a difference because the rules of interaction with the blockchain may or may not be the same. This is called blockchain governance. The public blockchains have a predefined set of rules and the private ones can have a different set of rules per blockchain. A private blockchain may have different governance rules. For example, the token, gas Price, transaction fee, endpoints, etc. may or may not be the same in the aforementioned private Ethereum ledger and the Ethereum main net. This can impact our applications too.

In our application, we primarily focused on the public test network of Ethereum called Sepolia Testnet. While the basic concepts of interacting with private deployments of these blockchains will still be the same, there will be differences in how we configure our code to point to the private networks.

Ethereum testnets are simulated environments that replicate the conditions of the main Ethereum network, providing a sandbox for developers to test new features, upgrades, and smart contracts without the risk associated with using real assets. These testnets are crucial for the development process, enabling the identification and resolution of errors in a safe environment before deployment on the mainnet. This reduces the risk of bugs leading to asset loss or other critical issues.

Currently, there are three main Ethereum testnets:

**Sepolia:**

- **Purpose**: Specializes in the testing of smart contracts and decentralized applications (Dapps).
- **Launch**: Initially launched in October 2021 as a proof-of-authority network, Sepolia has since transitioned to a proof-of-stake consensus, reflecting Ethereum's mainnet environment.
- **Recommendation**: Recommended for smart contract development due to its up-to-date reflection of the mainnet's operational dynamics.

**Holesky:**

- **Purpose**: Designed for testing staking, infrastructure, and protocol development.
- **Focus**: This testnet is used for more technical aspects of the Ethereum ecosystem, ensuring that staking mechanisms and infrastructural components function correctly before mainnet implementation.

**Goerli:**

- **Purpose**: Facilitates the testing of network upgrades and validator operations.
- **Status**: While currently live, the Ethereum Foundation advises against using Goerli as it is deprecated.

**Importance of Ethereum Testnets**

Testnets are vital for the Ethereum development process for several reasons:

Safe Testing Environment: Developers can experiment with new features and detect bugs without the risk of losing real assets.

- **Mainnet Simulation**: Testnets closely simulate the mainnet's conditions, providing an accurate testing ground for Dapps and smart contracts.
- **Cost-Effective**: Test ether used in testnets has no real-world value, making it cost-effective for extensive testing and experimentation.
- **Community Collaboration**: Testnets are open to the Ethereum community, fostering collaboration and collective problem-solving.

The Ethereum Sepolia testnet stands out as a pivotal platform for developers. Launched in October 2021 as a proof-of-authority network by Ethereum's core developers, Sepolia has since transitioned to a proof-of-stake (PoS) consensus, mirroring Ethereum's mainnet environment. This testnet is recommended for smart contract development due to its up-to-date reflection of the mainnet's operational dynamics.

**Key Points about the Ethereum Sepolia Testnet**

- **Purpose**: The primary purpose of the Sepolia testnet is to provide a platform for developers to test their projects in a controlled environment. This helps to ensure the stability and security of applications before they are deployed on the mainnet.
- **Compatibility**: Sepolia is designed to be compatible with the Ethereum mainnet, enabling developers to use the same tools, libraries, and protocols for both testing and deployment.
- **Network Parameters**: Sepolia has its own network parameters, such as block time, gas limits, and staking requirements. These parameters may differ from the mainnet and are often adjusted to simulate various network conditions for comprehensive testing.
- **Test Ether**: Sepolia uses a native crypto currency called "test ether" or "testnet ether" (ETH). This test ether has no real-world value and is used for transactions, deploying smart contracts, and interacting with Dapps on the testnet. It can be obtained from faucets or other sources specifically designed for testnet tokens.
- **Development and Testing**: Developers use Sepolia to experiment with new Ethereum features, conduct security audits, and identify potential vulnerabilities. This testing ensures that applications are robust and secure when they go live on the mainnet.

- **Community Participation**: The Sepolia testnet is open to anyone who wants to participate in testing or development. It is supported by the Ethereum community and various development tools like MetaMask, Remix, and Truffle.
- **Upgrades and Forks**: Like the mainnet, Sepolia may undergo upgrades or hard forks to implement new features or improvements. These changes are typically tested on Sepolia first to assess their impact before being applied to the mainnet.
- **Proof of Stake (PoS) Consensus Mechanism**: Following Ethereum's transition to Ethereum 2.0, Sepolia uses the Proof of Stake (PoS) consensus mechanism. PoS improves energy efficiency, security, and scalability compared to the previous Proof of Work (PoW) system.
- **Validator Selection**: In PoS, validators are chosen to propose and validate new blocks based on the number of tokens they hold and are willing to "stake" as collateral. This selection process replaces the competitive mining process used in PoW.
- **Block Validation**: Validators take turns proposing and validating blocks, ensuring transactions are processed efficiently. They are rewarded for their participation with transaction fees and, sometimes, additional tokens.
- **Security and Finality**: PoS provides increased security and faster finality for transactions, reducing the likelihood of forks and double-spending attacks. Validators who act maliciously or fail to perform their duties risk losing their staked tokens.

**Why Choose Sepolia?**

Sepolia distinguishes itself among Ethereum testnets with several key features that make it particularly advantageous for developers:

**Permissioned Validator Set:**

- **Streamlined Operation**: Sepolia operates with a permissioned validator set, ensuring a more controlled and predictable environment. This streamlining is beneficial for developers who require a stable and consistent testnet experience.

- **Storage Efficiency**: Running nodes on Sepolia is more storage-efficient, which is advantageous for developers who need to quickly sync with the network without the burden of extensive storage requirements.

**Uncapped Testnet ETH Supply:**

- **Ample Resources**: Unlike other Ethereum testnets, Sepolia does not limit the supply of testnet ETH. This ensures that developers always have sufficient resources for their development activities.
- **Continuous Development**: The unrestricted availability of testnet ETH allows for uninterrupted development and testing, facilitating a smoother and more productive workflow for developers working on smart contracts, Dapps, and other projects.

**Reflective of Mainnet Environment**:

- **Up-to-Date Dynamics**: Sepolia is designed to closely mirror the operational dynamics of the Ethereum mainnet. This makes it an ideal environment for smart contract development, as it provides developers with a realistic testing ground.
- **Consistent Upgrades**: Sepolia undergoes regular upgrades and forks, similar to the mainnet, ensuring that developers can test their applications in an environment that stays current with the latest Ethereum developments.

**Efficient for Node Operators:**

- **Quick Syncing**: The efficient design of Sepolia allows for faster syncing times, which is particularly beneficial for developers who need to get their nodes up and running quickly.
- **Resource Management**: By requiring less storage and computational power, Sepolia makes it easier for developers to manage their development resources effectively.

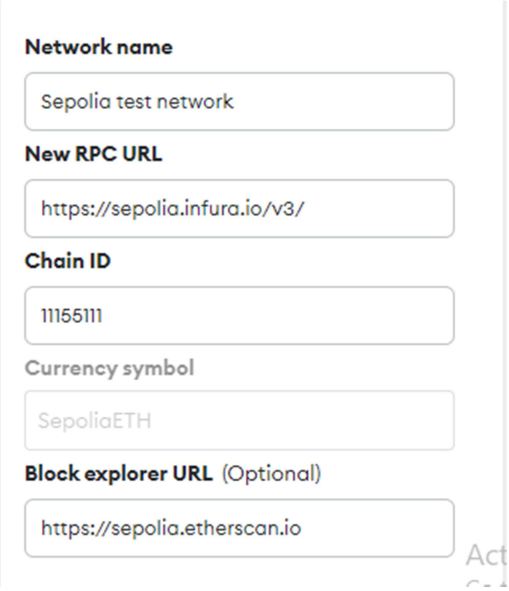
Sepolia was a proof-of-authority testnet created in October 2021 by Ethereum core developers and maintained ever since. After the Ropsten testnet reached a Terminal Total Difficulty (TTD) of 50000000000000000 the Sepolia and the Goerli testnets transitioned to a proof-of-stake consensus mechanism to mimic the Ethereum mainnet.

Testnets are blockchains designed to mimic the operating environment of a 'mainnet' but exist on a separate ledger. These testnets help developers test their applications and smart contracts in a risk-free way before deploying their products to Ethereum's mainnet environment.

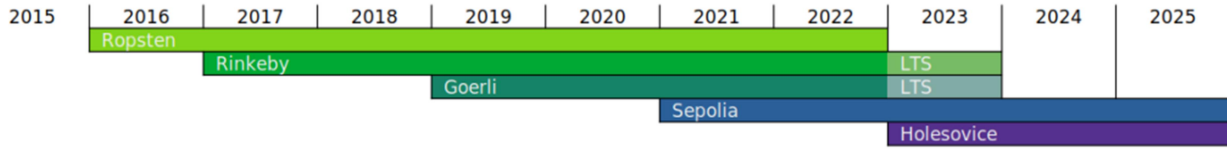
Sepolia was designed to simulate harsh network conditions, and has shorter block times, which enable faster transaction confirmation times and feedback for developers.

Compared to other testnets like Goerli, Sepolia's total number of testnet tokens is uncapped, which means it is less likely that developers using Sepolia will face testnet token scarcity like Goerli.

The Basic Information of Testnet used in this project




**Historical Ethereum Testnets**




**Source: GitHub ("Holesovice" renamed Holešky)**

**The following are some of the historical Ethereum testnets:**

**1. Olympic:** The Olympic testnet was launched in 2015 and was the first public Ethereum testnet. It was used to test the early versions of the Ethereum protocol and smart contract platform.

**2. Morden:** The Morden testnet was launched in 2016 and was a major improvement over the Olympic testnet. It was used to test the first versions of the Ethereum mainnet.

**3. Ropsten:** The Ropsten testnet was launched in 2016 and was one of the most popular Ethereum testnets for many years. It was used to test and deploy a wide variety of smart contracts and applications.

**4. Kovan:** The Kovan testnet was launched in 2017 and was another popular Ethereum testnet. It was used to test and deploy smart contracts and applications, as well as to test network upgrades.

**5. Kiln:** Kiln was a public Ethereum testnet that was designed to simulate the Ethereum mainnet's merge with the Beacon Chain, which was a key step in the transition to proof of stake. The Kiln testnet was successful in simulating the merge, and it provided a valuable environment for developers and node operators to test their applications and infrastructure before the merge took place.

**6. Rinkeby:** Rinkeby was a Proof-of-Authority testnet that provided efficient transactions and was a popular Ethereum testnet for development, with a host of active developers and projects using it. Rinkeby was deprecated in June 2023, and it will be shut down in Q2/Q3 2023.

| Testnet | Consensus Mechanism | Client Compatibility | Key Features | Reliability | Use Cases |
|---|---|---|---|---|---|
| Ropsten | Proof-of-Work (PoW) | Geth | Oldest Ethereum testnet, susceptible to spam attacks. | Occasionally experiences stability issues. | General smart contract testing and development. |
| Rinkeby | Proof-of-Authority (PoA) | Geth, Pantheon | Uses Clique consensus. Validators are block signers. | More stable due to PoA consensus. | Development, testing, and experimentation. |
| Goerli | Cross-Client, PoA | Geth, Pantheon | Cross-client compatibility, PoA consensus mechanism. | Highly reliable and resistant to spam attacks. | Cross-client interoperability testing. |
| Kovan | PoA | Parity | Uses the Parity Ethereum client with PoA consensus. | Reliable and widely used for testing. | General smart contract testing and development. |
| Görli | Cross-Client, PoA | Geth, Pantheon | PoA consensus, cross-client compatibility. | High reliability and spam-resistant. | Cross-client interoperability and general testing. |
| Morden (Legacy) | PoW | Geth | Deprecated, replaced by more modern testnets. | Not recommended for new developments. | Historical purposes, not recommended for active projects. |
| Ethereum 2.0 Testnets | PoS | Various | Testnets for Ethereum's transition to PoS consensus. | Varied, with occasional upgrades and resets. | Testing the features and dynamics of Ethereum 2.0 transition. |

**Why Proof of Stake?**

The Proof of Stake (PoS) algorithm is a consensus mechanism that has gained popularity for achieving distributed consensus in blockchain networks. Unlike Proof of Work (PoW), PoS is focused on validating blocks of transactions rather than mining new coins. In PoS systems, validators, often referred to as miners for simplicity, earn transaction fees instead of mining rewards. Validators must bond or mortgage their crypto currency stake to participate in the validation process. The likelihood of a validator being chosen to produce a new block is proportional to the amount of crypto currency they have staked; the more they stake, the greater their chance of validation. For instance, if a validator holds 2% of all Ether (ETH) in the Ethereum network, they could theoretically validate 2% of all transactions. The selection process for creating new blocks varies depending on the specific PoS algorithm, with notable examples including naive PoS, Delegated PoS (DPoS) used by BitShares, chain-based PoS, BFT-style PoS, and Casper PoS being developed for Ethereum.

In PoS systems, block creation is deterministic and based on the amount at stake, allowing for faster operation compared to PoW systems. Since there are no block rewards, the total amount

of digital currency is fixed and pre-distributed from the beginning. PoS can offer better protection against malicious attacks, as executing an attack would put the attacker's staked crypto currency at risk. Additionally, PoS is more energy-efficient because it does not require extensive computational power, making it a more environmentally friendly alternative to PoW where applicable.

**What are Decentralized Applications (DApps)?**

Decentralized applications Dapps are software programs that operate on decentralized networks such as blockchains or peer-to-peer P2P networks. Unlike traditional applications which rely on centralized servers and control Dapps leverage the decentralized nature of blockchain technology to provide enhanced security transparency and resilience.

**Key Characteristics of Dapps:**

- **Decentralization:** Dapps are hosted on a network of nodes rather than a single central server. This distribution means that no single entity has complete control over the application which reduces the risk of censorship and central points of failure. Each node in the network maintains a copy of the blockchain ledger which records all transactions and changes. This distributed ledger is updated through consensus mechanisms ensuring that all participants agree on the state of the blockchain.
- **Open Source:** Most Dapps are opensource meaning their source code is publicly accessible. This transparency allows anyone to review inspect and contribute to the code fostering trust and collaborative development. Opensource projects benefit from community scrutiny and input which can enhance security and innovation. By enabling communitydriven development opensource Dapps often evolve rapidly and incorporate diverse perspectives and improvements.
- **Smart Contracts:** Smart contracts are selfexecuting contracts with the terms of the agreement written directly into code. These contracts automatically enforce and execute the terms of the agreement without the need for intermediaries. Smart contracts facilitate trustless transactions meaning that parties do not need to trust each

other they only need to trust the code. This automation can reduce costs and increase efficiency by eliminating the need for manual enforcement and oversight.

- **Consensus Mechanisms:** Consensus mechanisms are protocols used to achieve agreement on the blockchains state across a distributed network. Examples include Proof of Work PoW and Proof of Stake PoS PoW requires participants miners to solve complex cryptographic puzzles to validate transactions and create new blocks while PoS involves participants validators who lock up a stake of cryptocurrency to earn the right to validate transactions. These mechanisms ensure that all nodes agree on the transaction history and prevent fraudulent activities.
- **Cryptographic Security:** Dapps utilize cryptographic techniques to secure data and transactions. Data on the blockchain is hashed using cryptographic algorithms which transforms it into a fixed size string of characters that is nearly impossible to reverse engineer. This ensures the immutability of the data as once information is written to the blockchain it cannot be altered without altering all subsequent blocks. Private keys are used by users to sign transactions and access their assets. The security of these private keys is crucial as loss or theft can result in irreversible asset loss.
- **Tokenization:** Tokenization involves creating digital tokens that represent assets or rights within a Dapp. These tokens can serve various purposes such as facilitating transactions granting access to specific features or participating in governance. Utility tokens are often used within the application to access services or features while governance tokens allow holders to vote on changes or upgrades to the Dapp. Tokenization can also enable fractional ownership of assets and create new economic models.

**Benefits of DApps:**

- **Cost and Efficiency:** By removing intermediaries like banks or payment processors Dapps can significantly reduce transaction fees and processing times. Traditional financial systems often involve multiple parties each taking a fee but Dapps streamline processes by executing transactions directly on the blockchain. Additionally the

automation provided by smart contracts reduces administrative overhead and operational costs making complex processes more efficient.

- **Security:** The immutability of blockchain technology provides a high level of security. Once data is recorded on the blockchain it cannot be altered or deleted which helps prevent fraud and unauthorized changes. The decentralized nature of Dapps means there is no single point of failure making them more resilient to attacks. Cryptographic techniques ensure that data is secure and transactions are validated correctly.
- **Accessibility:** Dapps are accessible from anywhere with an internet connection making them available to a global audience. This accessibility can promote financial inclusion by providing services to individuals who lack access to traditional banking systems. Dapps can also enable crossborder transactions and interactions without the need for intermediaries making them a valuable tool for users in regions with limited access to financial services.
- **Transparency:** Transactions and changes on the blockchain are recorded on a public ledger that is visible to all participants. This transparency allows users to verify the integrity of data and transactions without relying on centralized authorities. The ability to audit transactions in real time enhances accountability and trust within the system as users can independently verify that processes are being executed as intended.

**Drawbacks of DApps:**

- **Scalability Issues:** One of the main challenges faced by Dapps is scalability. Blockchain networks often have limitations on the number of transactions they can process per second. As the number of users and transactions increases the network can become congested leading to slower transaction times and higher fees. Solutions such as layer 2 scaling technologies and sharding are being explored to address these issues but scalability remains a significant challenge for many Dapps.
- **User Interface Challenges:** The user experience of Dapps can be more complex compared to traditional applications Interacting with blockchain technology often requires a certain level of technical knowledge which can be a barrier for new users.

Additionally integrating blockchain functionality into userfriendly interfaces can be challenging as it requires designing systems that are both intuitive and secure. Improving user interfaces and making them more accessible is an ongoing area of development for Dapps.

- **Development and Maintenance:** Developing and maintaining Dapps can be complex and resource intensive. Once a smart contract is deployed on the blockchain it is difficult to modify or correct which can be problematic if bugs or vulnerabilities are discovered. Extensive testing is required to ensure that smart contracts are secure and function as intended. Additionally the development process may involve coordinating with a decentralized community which can add complexity to decisionmaking and implementation.
- **Security Risks:** While blockchain technology provides a high level of security it is not immune to risks Smart contracts can contain vulnerabilities or bugs that may be exploited by malicious actors. Security breaches or exploits can lead to financial losses or data compromises Users must also manage their private keys securely as loss or theft of these keys can result in the irreversible loss of assets. Ensuring robust security measures and regular audits are essential for mitigating these risks.

**Centralized vs Decentralized Applications**

**Centralized Applications**

Centralized applications are controlled by a single organization or entity which owns and manages the application infrastructure. The application software resides on centralized servers controlled by the owner and all data is stored and managed by this central authority Users interact with the application by sending data to and from the companys servers. Centralized applications generally have more control over their infrastructure and can scale more easily due to centralized management.

**Decentralized Applications**

Dapps operate on blockchain or P2P networks with control distributed across a network of nodes. Data is distributed and replicated across the network enhancing security and reducing

the risk of single points of failure. Trust is established through cryptographic methods and consensus mechanisms rather than a central authority. While decentralized applications offer increased security and transparency they may face challenges related to scalability and user experience due to the complexities of distributed systems.

**Popular Dapps and Their Use Cases:**

- **Cryptocurrency Wallets:** MetaMask is a leading Dapp that serves as a browser extension and mobile app allowing users to manage their cryptocurrency assets and interact with Ethereum based applications. It provides a secure way to store private keys and facilitates transactions with various blockchain based services. MetaMask supports a wide range of cryptocurrencies and offers users an interface to connect with decentralized applications directly.
- **Decentralized Exchanges DEXs:** Uniswap is one of the most popular decentralized exchanges that enables users to trade cryptocurrencies directly from their wallets. It operates using an automated market maker AMM model which facilitates trading without the need for a centralized order book. SushiSwap another prominent DEX offers similar functionality but with additional features such as yield farming and staking opportunities.
- **NFT Marketplaces:** OpenSea is a prominent decentralized marketplace for buying selling and trading nonfungible tokens NFTs. It provides a platform for users to create auction and trade digital assets like artwork collectibles and virtual items. OpenSea supports a wide range of NFTs and has become a significant player in the digital collectibles market.
- **Gambling Dapps:** MetaWin is a decentralized gambling Dapp that allows users to participate in various betting games and lotteries Leveraging blockchain technology. MetaWin ensures transparency and fairness in gambling activities by recording all transactions and outcomes on the blockchain. This transparency helps build trust among users and reduces the risk of fraud.
- **Decentralized Finance DeFi:** Aave is a decentralized lending and borrowing platform that enables users to earn interest on deposits and take out loans without

intermediaries. It uses smart contracts to automate the lending process and manage collateral. Compound is another DeFi platform that provides similar services allowing users to lend and borrow cryptocurrencies in a decentralized manner.

- **Gaming Dapps:** Axie Infinity is a blockchainbased game where players can collect breed and battle fantasy creatures called Axies. It integrates NFT technology to provide true ownership of ingame assets allowing players to buy sell and trade these assets within the game. Axie Infinity has gained popularity for its playtoearn model which rewards players with cryptocurrency.
- **Social Media Platforms:** Steemit is a decentralized social media platform that rewards users with cryptocurrency for creating and curating content. Unlike traditional social media networks Steemit allows users to monetize their content and participate in community governance. The platform emphasizes content creation and engagement with users earning rewards based on their contributions.
- **Prediction Markets:** Augur is a decentralized prediction market platform that allows users to create and trade prediction markets on various events. Participants can place bets on the outcome of future events and the platform uses blockchain technology to ensure market integrity and transparency. Augur enables users to speculate on a wide range of topics from political events to sports outcomes.

**Architecture of a Dapp**

The architecture of Dapp or Decentralized applications is quite different from traditional applications. Decentralized applications require a unique system design to achieve high security, reliability, and privacy.

In decentralized applications, there's no centralized database that stores the application state. It means, unlike centralized applications the client-side application does not communicate to the database instead it communicates directly to the blockchain.

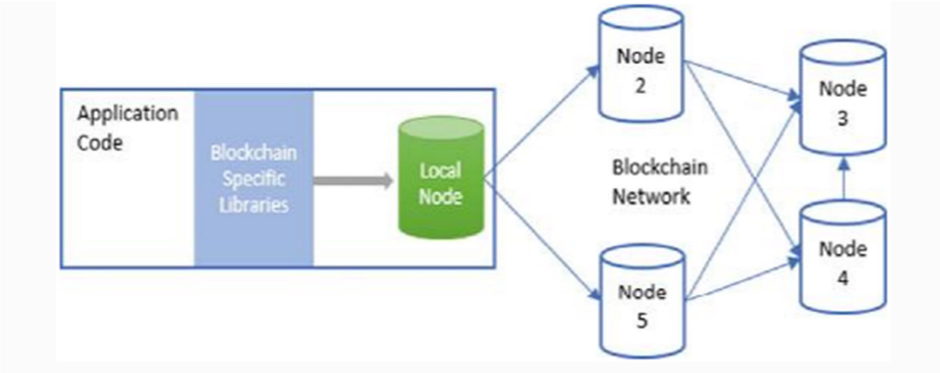
The major components of the Dapp architecture are Ethereum blockchain, Smart Contracts, Ethereum Virtual Machine (EVM) and Front-end.

**Communication Between Frontend And Smart Contract On Ethereum**

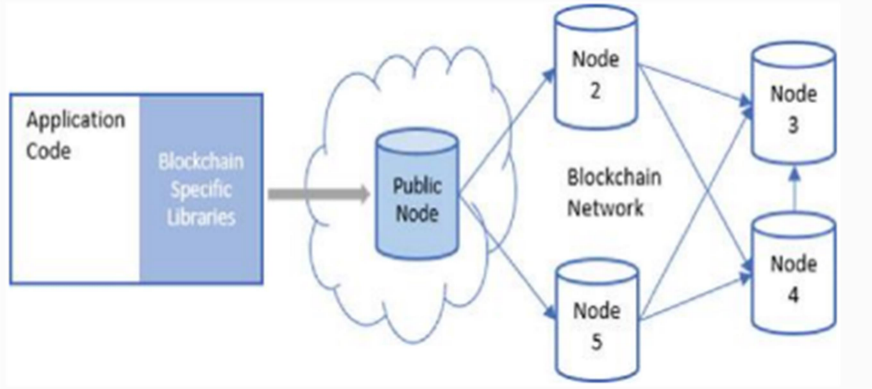
To invoke functions the front end needs to communicate with smart contracts. But Ethereum is a decentralized network. Every node in the Ethereum network keeps a copy of all states on the Ethereum Virtual machine, including the data and code associated with every smart contract.

It is required to interact with one of these nodes to interact with the data and the code on the blockchain. This is because any node can broadcast a request for a transaction to be executed on the EVM. Then the job of a miner is to execute the transaction and propagate the resulting state change to the network. The transaction can be broadcasted in two ways:

Application and node both run locally: The application and the node both run on the local machine. This means we will need our application users to run a local blockchain node and point the application to connect with it. This model would be a purely decentralized model of running an application. An example of this model is the Ethereum-based Mist browser, which uses a local geth node.



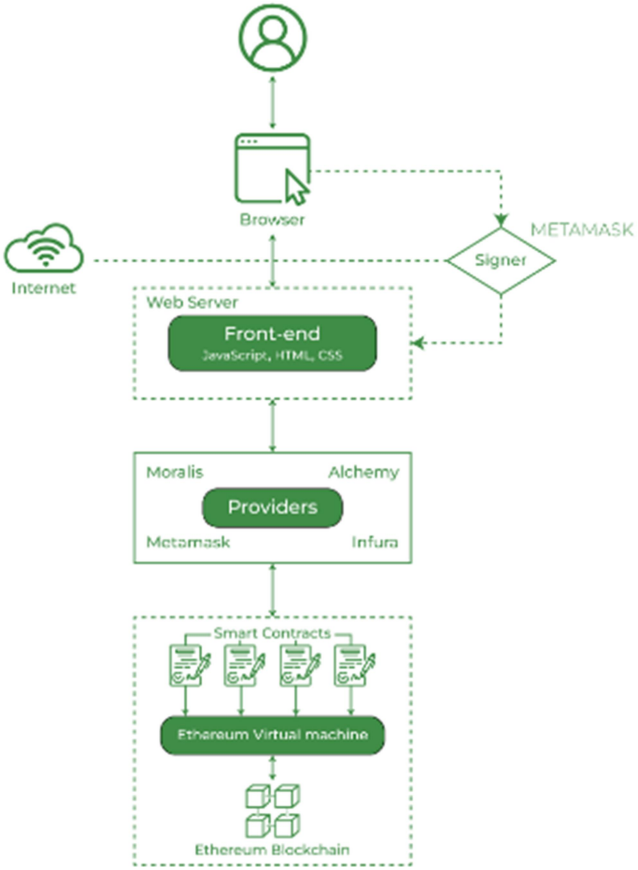

**Public node:** The application talks to a public node hosted by a third party. This way our users don't have to host a local node. There are several advantages and disadvantages of this approach. While the users don't have to pay for power and storage for running a local node, they need to trust a third party to broadcast their transactions to the blockchain. The Ethereum browser plugin metamask uses this model and connects with public hosted Ethereum nodes.




**Why Use Third-party Services?**

When third-party services are used there is no need to run a full node yourself. Setting up a new Ethereum node on the server can take much time. Moreover, more nodes need to be added to expand your infrastructure and the cost of storing the full Ethereum blockchain goes up as Dapp scales. The nodes one connects with to interact with the blockchain are often called "providers."

Ethereum provider (i.e. nodes) implements a JSON-RPC specification. This ensures that there's a uniform set of methods when our frontend applications want to interact with the blockchain.

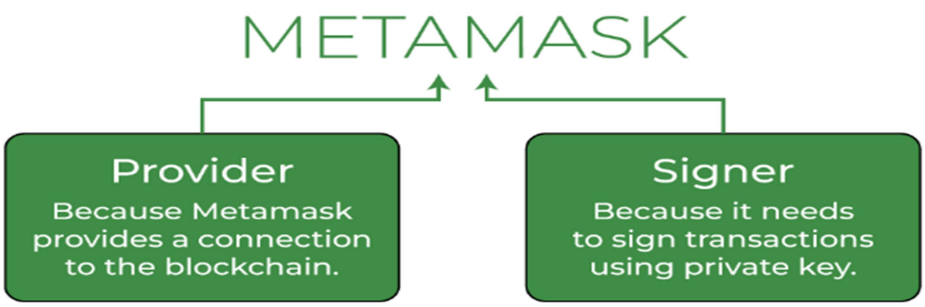



One can read the state stored on the blockchain once connected to the blockchain through a provider. But if one wants to write to the state before one can submit the transaction to the blockchain there's one more thing that needs to be done and the thing is "sign" the transaction using the private key.

Whenever the front end needs the user to sign a transaction, it calls on Metamask. Here Metamask plays the role of the signer.

**The role of Metamask in Dapp Architecture:-**

Metamask is a browser plugin that serves as an Ethereum wallet. we know that the Ethereum blockchain is a network and we need a special browser extension to connect that network.




Metamask will allow us to connect the blockchain with our personal account and interact with the smart contract.

A user's private keys are stored by Metamask in the browser, and when a user wants to write to the state, the user needs to "sign" the transaction using the private key.

**Metamask**

Metamask acts both as a provider and a signer because Metamask provides a connection to the blockchain (as a "provider") and since it needs it to sign transactions that's why it is also a signer.

**Why we need IPFS?**

IPFS (InterPlanetary File System) enables a decentralized, peer-to-peer approach to data storage and sharing, improving resilience, security, and efficiency. By using content addressing and distributing data across multiple nodes, it reduces dependency on central servers, enhances data integrity, and supports faster, more reliable access to information.

In blockchain Dapps, IPFS is used to store and share large files off-chain while maintaining a decentralized and secure reference on-chain. This approach leverages IPFS's distributed storage to improve scalability and performance, while blockchain technology ensures data integrity and decentralized control.

## Literature Review

The reviewed literature emphasizes blockchain technology's transformative potential in enhancing academic credential verification and security. Key findings highlight blockchain's capacity to revolutionize the verification process by addressing challenges like forged certificates, manual verification, and data sharing. Key components include distributed ledgers, cryptography, smart contracts, and decentralized storage, offering increased security, efficiency, and transparency. Widespread adoption of blockchain faces challenges such as scalability limitations, privacy concerns, regulatory gaps, automation issues, smart contract immutability, maintenance costs, and energy consumption. Despite these challenges, blockchain holds the promise to revolutionize academic certificate issuance and verification, enhancing overall security, efficiency, and trustworthiness. Numerous studies propose blockchain-based solutions, emphasizing benefits like immutability, traceability, and decentralization in academic certificate verification.

A number of studies have been conducted in India and abroad on various aspects of blockchain technology, especially in academic credential verification and security. Following are the few systems which were explained by the researchers in their research study:

**Aamna Tariq, Hina Binte Haq, Syed Taha Ali (2019)**, in their research study, "**Cerberus: A Blockchain-Based Accreditation and Degree Verification System**", discussed the prevalent issue of credential fraud in higher education systems, emphasizing its negative impact on investment and confidence. They critiqued traditional verification systems as being time-consuming and ineffective against certain types of fraud. Their proposed solution, Cerberus, was a blockchain-based credential verification system designed to be more efficient, user-friendly, and capable of addressing various forms of fraud. Unlike other blockchain solutions, Cerberus aligned closely with existing verification practices, considered real-world fraud scenarios, and employed on-chain smart contracts for credential revocation. Importantly, it did not require users to manage digital identities or cryptographic credentials. The authors believed Cerberus could positively contribute to combating fake credentials and hoped it would be

adopted by universities, accreditation bodies, and employers to enhance the verification process and customize features according to their needs. The managerial relevance statement highlighted the potential of blockchain to combat fake credentials and emphasized the need for solutions tailored to existing fraud practices and user usability concerns. The authors encouraged practitioners to visualize and integrate Cerberus into their credential verification systems, adapting its features to meet their specific requirements. The authors presented Cerberus as a promising solution to address the widespread issue of credential fraud in a practical and user-friendly manner. [3]

**Leka, Elva & Selimi, Besnik (2020)** in their paper **"BCERT - A Decentralized Academic Certificate System Distribution Using Blockchain Technology"** proposes a blockchain-based system, BCert, for storing, distributing, and verifying academic certificates to enhance efficiency and security. BCert utilizes Ethereum smart contracts and IPFS for decentralized file storage. Academic certificates are hashed and stored on a public blockchain, ensuring validity through cryptographic signatures. The system aims to provide confidentiality by encrypting data with the AES algorithm. BCert covers blockchain components such as traceability, provenance, certification, and authentication, reducing transaction and smart contract deployment costs. Tests show a cost of approximately $2 for transacting 170 bytes of data. The system benefits stakeholders such as educational institutions, government, labor market, and professionals, offering real-time verification, third-party verification, usability, and fraud reduction. The authors emphasizes potential benefits for education quality, labor market efficiency, and fraud prevention through widespread blockchain adoption. [4]

**Ayub Khan, Abdullah, Asif Ali Laghari, Aftab Ahmed Shaikh, Sami Bourouis, Amir Madany Mamlouk, and Hammam Alshazly (2021)** in their paper **"BCERT - A Decentralized Academic Certificate System Distribution Using Blockchain Technology"** proposes a blockchain-based system, BCert, for storing, distributing, and verifying academic certificates to enhance efficiency and security. BCert utilizes Ethereum smart contracts and IPFS for decentralized file storage. Academic certificates are hashed and stored on a public blockchain, ensuring validity through

cryptographic signatures. The system aims to provide confidentiality by encrypting data with the AES algorithm. BCert covers blockchain components such as traceability, provenance, certification, and authentication, reducing transaction and smart contract deployment costs. Tests show a cost of approximately $2 for transacting 170 bytes of data. The system benefits stakeholders such as educational institutions, government, labor market, and professionals, offering real-time verification, third-party verification, usability, and fraud reduction. The authors emphasizes potential benefits for education quality, labor market efficiency, and fraud prevention through widespread blockchain adoption. [5]

**Kistaubayev, Yerlan, Galimkair Mutanov, Madina Mansurova, Zhanna Saxenbayeva, and Yassynzhan Shakan (2022)** in their research article on **"Ethereum- Based Information System for Digital Higher Education Registry and Verification of Student Achievement Documents**" discussed the integration of blockchain technology, specifically based on the Ethereum blockchain architecture, in the field of education to address challenges related to the verification of academic achievements and the security of educational documents. Their main focus was on developing a platform called UniverCert, which created a unified digital register of students & educational achievements. The authors emphasized the importance of considering performance criteria, such as throughput, transaction speed, and data storage in the Ethereum blockchain, in evaluating the feasibility of the proposed solution. The UniverCert solution utilized a consortium architecture with a single Progress smart contract, optimizing transaction costs by storing basic student data off-chain and using identifiers in blockchain transactions. Their paper concluded that the developed blockchain solution was applicable for accounting and verifying student academic achievements, providing potential advantages for the higher education system of the Republic of Kazakhstan with minimal costs. Authors aimed to enhance data integrity, security, and authenticity in the context of digitizing and automating educational management processes. [6]

**Rahman, Tasfia, Sumaiya Islam Mouno, Arunangshu Mojumder Raatul, Abul Kalam Al Azad, and Nafees Mansoor (2023)** in their paper **"Verifi-Chain: A Credentials Verifier using**

**Blockchain and IPFS"** proposed a solution to the common problem of fake certificates in Southeast Asia, which hindered qualified candidates from securing deserving jobs. The proposed solution involved leveraging blockchain technology and the Interplanetary File System (IPFS) to create a secure and tamper-proof system for verifying academic credentials. In their suggested prototype, certificates were temporarily stored in a database before being transferred to IPFS, where a unique hash code was generated and stored in the blockchain nodes. This hash code served as the certificate's unique identity. Their system aimed to make the certificate verification process more efficient, secure, and cost-effective for companies by allowing them to easily access and verify already authenticated certificates. The use of IPFS as an intermediary storage platform was highlighted as a cost-saving measure compared to directly storing massive data on the blockchain. The authors emphasized the advantages of blockchain, such as creating immutable ledgers, and envisioned the proposed solution revolutionizing the verification of academic certificates, making it more transparent and secure for all stakeholders. Additionally, their research suggested that future work could explore integrating other blockchain platforms and file storage systems to enhance the system and functionality and robustness. [7]

**K. V Raghavender, S. Alankruthi, A. Akhila, T. Preethi, and M. Ashritha (2023)** proposed a solution in their paper **"Decentralized Smart Contract Certificate System Using Ethereum Blockchain Technology**" to address the challenges associated with traditional paper and electronic certificates by leveraging blockchain technology for certificate verification. In this proposed system, universities uploaded student certificates to a federated blockchain, generating a unique hash on the Interplanetary File System (IPFS). This allowed anyone, including students, recruiters, or administrators, to verify certificates by providing a unique hash and roll number. The use of blockchain ensured security, reduced costs, and expedited the verification process. The suggested future directions included expanding the system for document authentication in various sectors, eliminating fraudulent certificates, and enhancing usability features. [8]

**Shaik Arshiya Sultana, Chiramdasu Rupa, Ramanadham Pavana Malleswari, and Thippa Reddy Gadekallu (2023)** proposed a solution to the challenges of verifying academic certificates in the digital age by leveraging blockchain technology in their study, **"IPFS-Blockchain Smart Contracts Based Conceptual Framework to Reduce Certificate Frauds in the Academic Field"**. Their proposed solution utilizes blockchain technology to store academic certificates as secure digital assets, ensuring their integrity and immutability through cryptographic measures such as unique identifiers and timestamps. This approach eliminates the need for intermediaries and reduces the risk of fraudulent credentials, as verification can be performed directly on the decentralized Ethereum network using MetaMask. The study emphasizes the importance of using unique identifiers for encrypted data stored on the blockchain, with additional security provided by encryption techniques. The decentralized nature of the blockchain system, coupled with encryption, enhances security and makes fraudulent activities, such as forging or tampering with certificates, highly challenging.

Furthermore, the proposed system offers improved efficiency in seamlessly verifying academic certificates, saving time and resources for employers and institutions while uilding trust in the authenticity of credentials. The proposed solution represents a significant step forward in addressing the challenges of academic credential verification in the digital age. By harnessing the power of blockchain technology and cryptographic measures, the system has the potential to revolutionize credential management within the education sector, fostering greater trust, transparency, and efficiency. [9]

There are a number of different blockchain-based solutions for verifying academic certificates in development. These solutions are still in their early stages, but they have the potential to make a significant impact on the education sector.

## Problem Definition

The current system for verifying academic credentials is plagued by numerous challenges that impact security, authenticity, and efficiency. These traditional methods, which often involve centralized databases, paper certificates, and manual verification processes, are increasingly seen as inadequate. This deficiency has sparked a pressing demand for more secure, reliable, and efficient solutions to enhance the integrity of academic credentials. Below is a comprehensive breakdown of the core issues:

**1. Security Risks**

Traditional methods of storing and verifying academic credentials are highly susceptible to various security threats. The reliance on centralized databases creates a single point of failure, making them attractive targets for cybercriminals. Data breaches can lead to unauthorized access to sensitive personal information, resulting in identity theft and fraudulent activities, such as the issuance of fake degrees. Additionally, without robust encryption and security protocols, the risk of tampering with academic records is significantly elevated.

**2. Lack of Transparency**

The current verification process often lacks transparency, making it difficult for stakeholders—including employers, educational institutions, and regulatory bodies—to efficiently authenticate academic credentials. This opacity can lead to skepticism regarding the legitimacy of qualifications, hindering the trust between graduates and employers. Without clear, verifiable methods of credential validation, stakeholders are left to navigate a complex landscape fraught with uncertainty.

**3. Time-Consuming Verification Process**

The manual verification of academic credentials is an arduous and resource-intensive task. It typically involves contacting multiple institutions, verifying records, and cross-checking information, which can take days or even weeks to complete. This not only delays recruitment processes but also places an administrative burden on educational institutions, which must

allocate resources to manage the influx of verification requests. As a result, valuable time is lost in both hiring and administrative cycles.

**4. Potential for Credential Fraud**

The advent of sophisticated printing and digital manipulation technologies has made it easier for individuals to produce counterfeit academic credentials. Fake degrees, diplomas, and transcripts are readily available for purchase online, undermining the credibility of genuine academic qualifications. This proliferation of fraudulent credentials can lead to misrepresentation in the workforce, eroding trust in educational institutions and compromising the integrity of the academic system.

**5. Inefficient Record-Keeping Systems**

Many educational institutions still rely on outdated and inefficient record-keeping systems that are prone to errors and inconsistencies. These systems often lack automation and fail to provide real-time updates on academic achievements, leading to challenges in maintaining accurate and up-to-date records. Data loss can occur due to inadequate backup procedures, further complicating the verification process and hindering the ability to track students' academic progress effectively.

## Problem Solution

Implementing blockchain technology can revolutionize the way academic credentials are stored, verified, and shared. Blockchain offers a decentralized and immutable ledger system that ensures the security, integrity, and transparency of academic credentials.

**Decentralized Storage:** Academic credentials can be securely stored on a blockchain network, eliminating the need for centralized repositories susceptible to hacking or tampering. Each credential is stored as a unique digital asset, cryptographically secured and accessible only through authorized channels.

**Immutable Records**: Once recorded on the blockchain, academic credentials cannot be altered or deleted, ensuring the integrity and authenticity of each record. This prevents fraudulent activities such as credential manipulation or falsification.

**Verification Process**: Employers, academic institutions, and other stakeholders can easily verify the authenticity of academic credentials by accessing the blockchain network. Verification can be done in real-time, eliminating the need for lengthy verification processes and reducing administrative burdens.

**Enhanced Privacy and Security**: Blockchain technology employs cryptographic techniques to ensure the privacy and security of sensitive data. Users have control over their credentials and can selectively share them with authorized parties while maintaining confidentiality.

Smart Contracts: Smart contracts can automate the verification and authentication process, allowing for seamless and transparent credential verification. Smart contracts can also facilitate the issuance, transfer, and revocation of credentials based on predefined conditions.

**Interoperability**: Blockchain standards and protocols enable interoperability among different educational institutions, employers, and credential verification platforms. This facilitates the seamless exchange and verification of academic credentials across diverse ecosystems.

**Reduced Costs and Efficiencies**: By eliminating intermediaries and streamlining verification processes, blockchain technology can significantly reduce costs and administrative overhead associated with credential verification. This makes the verification process faster, more reliable, and cost-effective for all stakeholders.

**User Empowerment**: Blockchain technology empowers individuals to have greater control and ownership over their academic credentials. They can easily access, manage, and share their credentials with confidence, knowing that they are secure and verifiable.

## Process Methodology and Approach

At foundation level following data will be collected for the system

**Credential Information**:

- Type: Diploma, certificate, or transcript.
- Identifier: Unique credential ID.
- Issuer: Institution name and contact.
- Issue Date: Date of issuance.
- Recipient: Student's name and contact.
- Credential Metadata: Program, major, graduation date.
- Credential Hash: Cryptographic hash ensuring the credential's integrity.

**Issuing Institution Information**:

- Name, ID, address, accreditation status and other required data.
- Authorized Representatives: Contact for verification.

**Student or Credential Holder Information:**

- Name, ID, contact, and other required data.

**Steps in credential issuance**

Educational institutions issue digital credentials to students upon completion of their studies. These credentials include the student's name, degree, date of graduation, and other relevant information.

The digital credentials are then hashed and stored on the IPFS. The hash generated by IPFS is then stored on blockchain. The hash is a unique fingerprint of the credential that cannot be altered.

The educational institution provides the transaction hash to their students. This transaction hash ensures that the credential has not been tampered with.

The student receives a copy of the digital credential and the hash.

**Steps in credential verification**

A potential employer or other stakeholder wants to verify the authenticity of a student's academic credentials.

The student provides the employer with the hash of their credential.

The employer can then search the blockchain for the hash and retrieve the corresponding credential.

The employer can verify the authenticity of the credential by comparing the hash of the credential on the blockchain to the hash provided by the student.

If the hashes match, then the employer can be confident that the credential is authentic.

Also, students are allowed to retrieve their credentials even though they lost their transaction hash. For this they need to provide their registered email or phone number for that credential and the name of the credential they want to retrieve. The OTP will be send on the entered email or phone number and if the OTP matches then the transaction hash will be auto filled in the input field. Further, students can go with verify, retrieve options or may provide that transaction hash to employer or any other stakeholders.

**Proposed System**

The fundamental mechanism of our proposed system revolves around the use of blockchain technology, specifically Ethereum Sepolia Testnet blockchain, a decentralized and immutable ledger. Ethereum Sepolia Testnet blockchain technology, connected via Infura API key, ensures seamless transaction management and data integrity.

**Code for connecting to Ethereum Sepolia Testnet using Infura API key:**

Connection.py

```
import tkinter as tk
from tkinter import messagebox
import socket
from web3 import Web3

infura_api_key = "64030d0ca36640db87457233878813"
ethereum_sepolia_rpc_url = "https://sepolia.infura.io/v3/"
# Connect to an Ethereum Sepolia Testnet node
w3 = Web3(Web3.HTTPProvider(ethereum_sepolia_rpc_url + infura_api_key))

def is_internet_available():
    try:
        socket.create_connection(("www.google.com", 80))
        messagebox.showinfo("Internet Connection", "Internet Connection Available!!")
        return True
    except OSError:
        messagebox.showinfo("Internet Connection", "No Internet Connection!! Please enable internet on your device")
        return False

def is_testnet_connected():
    if is_internet_available():
        try:
            if w3.is_connected():
                messagebox.showinfo("Testnet Connection", "Successfully connected to testnet")
                return True
            else:
                messagebox.showinfo("Testnet Connection", "Connection Failed! Try Again")
                return False
        except Exception as e:
            messagebox.showinfo("Testnet Connection", f"Connection Failed! Error: {e}")
            return False
    else:
        messagebox.showinfo("Internet Connection", "No Internet Connection!! Please enable internet on your device")
        return False
```

Each student's academic credentials are stored on IPFS using Pinata API and the hash generated of credentials are stored in a block, generating a unique Hash number, serving as the primary key. This pioneering approach facilitates seamless certificate verification for students and empowers employers to validate the authenticity of provided certificates.

**Code of uploading files to IPFS using Pinata API**

UploadOperation.py

```
import requests
import api.PinataCredentials

# Function to upload file to IPFS via Pinata
def upload_to_ipfs(file_path):
    files = {'file': open(file_path, 'rb')}
    response = requests.post('https://api.pinata.cloud/pinning/pinFileToIPFS',
                             files=files, headers=api.PinataCredentials.credential)
    if response.status_code == 200:
        return response.json()['IpfsHash']
    else:
        return None
```

**Code of Retrieving files from IPFS using Pinata API**

RetrieveOperation.py

```
import requests
import api.FileExistenceChecker

def retrieve_from_ipfs(ipfs_hash):

    if api.FileExistenceChecker.check_file_existence(ipfs_hash):
        response = requests.get(f"https://gateway.pinata.cloud/ipfs/{ipfs_hash}")
        if response.status_code == 200:
            file_name = response.url.split('/')[-1]
            return response.content
    else:
        return None
```

**Code of checking files presence in IPFS using Pinata API**

FileExistenceChecker.py

```
import requests
import api.PinataCredentials

def check_file_existence(file_hash):

    response = requests.get("https://api.pinata.cloud/data/pinList?status=pinned",
                            headers=api.PinataCredentials.credential)
    if response.status_code == 200:
        data = response.json()
        # Iterate through pinned items to check if file exists
        for item in data['rows']:
            if item['ipfs_pin_hash'] == file_hash:
                return True
        return False
    else:
        return False
```

Adding a new certificate to the blockchain incurs a nominal Ethereum gas fee. This certificate or any document of students will be saved on ipfs platform, which will generate the ipfs hash of file and this hash will be stored on the testnet.

This fee is deducted from the certificate authority's account and is essential for compensating miners who validate and add blocks to the blockchain.

For adding and retrieving the data stored on blockchain we have used smart contract.

**Code of Solidity Smart Contract compiled on Remix IDE and deployed on Blockchain**

UserDataStorage.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract UserDataStorage {
    event DataStored(address indexed user, string data);

    mapping(address => string) private userData;

    function storeData(string memory _data) public {
        userData[msg.sender] = _data;
        emit DataStored(msg.sender, _data);
    }

    function retrieveData(address user) public view returns (string memory) {
        return userData[user];
    }
}
```

**Function to deploy Smart Contract**

```
# Deploy the contract
def deploy_contract(deployer_address,deployer_private_key):
    UserDataStorage = con.w3.eth.contract(abi=abi, bytecode=bytecode)

    # Build transaction
    construct_txn = UserDataStorage.constructor().build_transaction({
        'from': deployer_address,
        'nonce': con.w3.eth.get_transaction_count(deployer_address),
        'gas': 6721975,
        'gasPrice': con.w3.to_wei('21', 'gwei')
    })

    # Sign transaction
    signed_txn = con.w3.eth.account.sign_transaction(construct_txn, private_key=deployer_private_key)
    # Send transaction
    tx_hash = con.w3.eth.send_raw_transaction(signed_txn.rawTransaction)
    # Wait for the transaction to be mined
    receipt = con.w3.eth.wait_for_transaction_receipt(tx_hash)
    # Print the deployed contract address
    contract_address = receipt.contractAddress
    return contract_address
```

**Function to store data on blockchain**

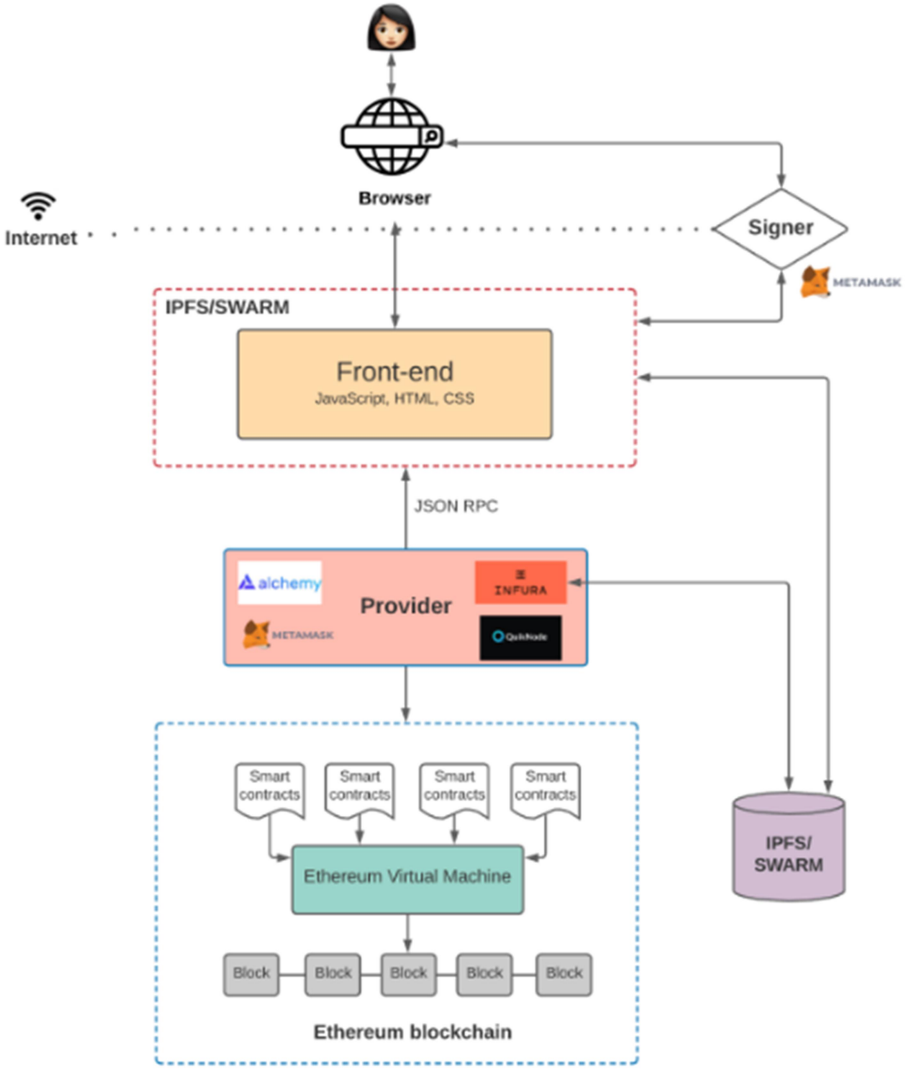
```
# Store data for a user
def store_data(contract, address, private_key, data):
    if con.is_internet_available():
        contract_instance = con.w3.eth.contract(address=contract, abi=abi)
        # Build transaction
        txn_dict = contract_instance.functions.storeData(data).build_transaction({
            'from': address,
            'nonce': con.w3.eth.get_transaction_count(address),
            'gas': 2000000,
            'gasPrice': con.w3.to_wei('10', 'gwei')
        })
        # Sign transaction
        signed_txn = con.w3.eth.account.sign_transaction(txn_dict, private_key=private_key)
        # Send transaction
        tx_hash = con.w3.eth.send_raw_transaction(signed_txn.rawTransaction)
        return tx_hash.hex()
    else:
        return None
```

**Function to retrieve data from blockchain**

```
# Retrieve data using event logs
def retrieve_data(contract, tx_hash):
    if con.is_internet_available():
        try:
            contract_instance = con.w3.eth.contract(address=contract, abi=abi)

            # Retrieve transaction receipt
            receipt = con.w3.eth.get_transaction_receipt(tx_hash)

            # Retrieve event logs
            logs = contract_instance.events.DataStored().process_receipt(receipt)

            # Print retrieved data
            if logs:
                user = logs[0]['args']['user']
                stored_data = logs[0]['args']['data']
                return True, user, stored_data
            else:
                return False, None, None
        except Exception as e:
            return False, None, None
    else:
        return False, None, None
```

To enhance the verification process, every student is assigned a unique Hash Id. This identifier provides a consolidated view of all associated certificates, streamlining the verification process and offering verifiers a comprehensive perspective. Since, we now know that, upon the addition of a certificate, a nominal Ethereum gas fee is incurred debited from the certificate authority's account. This fee is crucial for compensating miners, essential contributors responsible for adding blocks to the blockchain. In return, miners are rewarded with Ethereum

coins, ensuring the sustainability and integrity of the blockchain network. The distributed nature of the blockchain establishes a formidable defence against tampering. While acknowledging that absolute immunity is unattainable, the increasing length of the blockchain significantly elevates the difficulty of unauthorized data modification, providing a robust foundation for secure data storage. The proposed system serves as a pivotal link between educational institutions and industries. Institutions can securely store candidates' academic credentials, and industries can effortlessly verify these credentials using Hash number.




The proposed system also serves as a pivotal link between educational institutions and industries. Academic institutions can securely store student credentials on this platform, and industries can effortlessly verify them using Hash numbers.

### Code to read file contents

ReadFileData.py

```
import requests
import api.FileExistenceChecker

def read_ipfs_content(ipfs_hash):
    if api.FileExistenceChecker.check_file_existence(ipfs_hash):
        response = requests.get(f"https://gateway.pinata.cloud/ipfs/{ipfs_hash}")
        if response.status_code == 200:
            return response.content
        else:
            return None
    else:
        return None
```

### Code to Verify the file contents

VerifyOperation.py

```
import api.RetrieveOperation
import api.HashGenerator
import hashlib

def verify_file_contents(ipfs_hash, local_file_path):

    # First calculate the hash of the local file
    local_file_hash = api.HashGenerator.calculate_hash(local_file_path)

    # Second retrieve file content from ipfs
    ipfs_file_content = api.RetrieveOperation.retrieve_from_ipfs(ipfs_hash)

    if ipfs_file_content is None:
        print("Error retrieving file from IPFS.")
        return

    # Third calculate the hash of the content fetched from IPFS
    ipfs_hash_calculated = api.HashGenerator.calculate_ipfs_hash(ipfs_file_content)

    # Fourth compare the hashes
    if ipfs_hash_calculated == local_file_hash:
        return True
    else:
        return False
```

**Functions for OTP Generation and Verification**

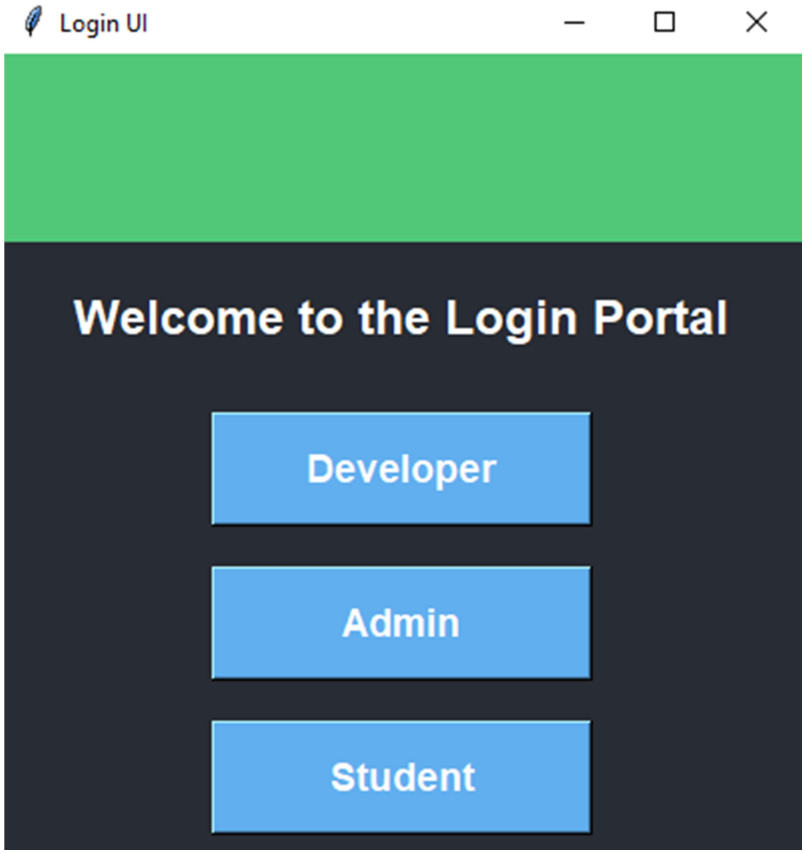
```
def generate_otp():
    global generated_otp, otp_timestamp
    generated_otp = "".join([str(random.randint(0, 9)) for _ in range(6)])
    otp_timestamp = time.time()
    return generated_otp

def send_otp(sender_email, sender_password, recipient_email, subject='Your OTP'):
    try:
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(sender_email, sender_password)

        otp = generate_otp()

        msg = MIMEMultipart()
        msg['From'] = sender_email
        msg['To'] = recipient_email
        msg['Subject'] = subject

        body = f'Your OTP is: {otp}'
        msg.attach(MIMEText(body, 'plain'))
        msg_string = msg.as_string()

        server.sendmail(sender_email, recipient_email, msg_string)
        server.quit()

        return otp
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

def validate_otp(input_otp):
    current_time = time.time()
    if generated_otp is not None and (current_time - otp_timestamp) <= 60:
        if generated_otp == input_otp:
            return True
    return False
```

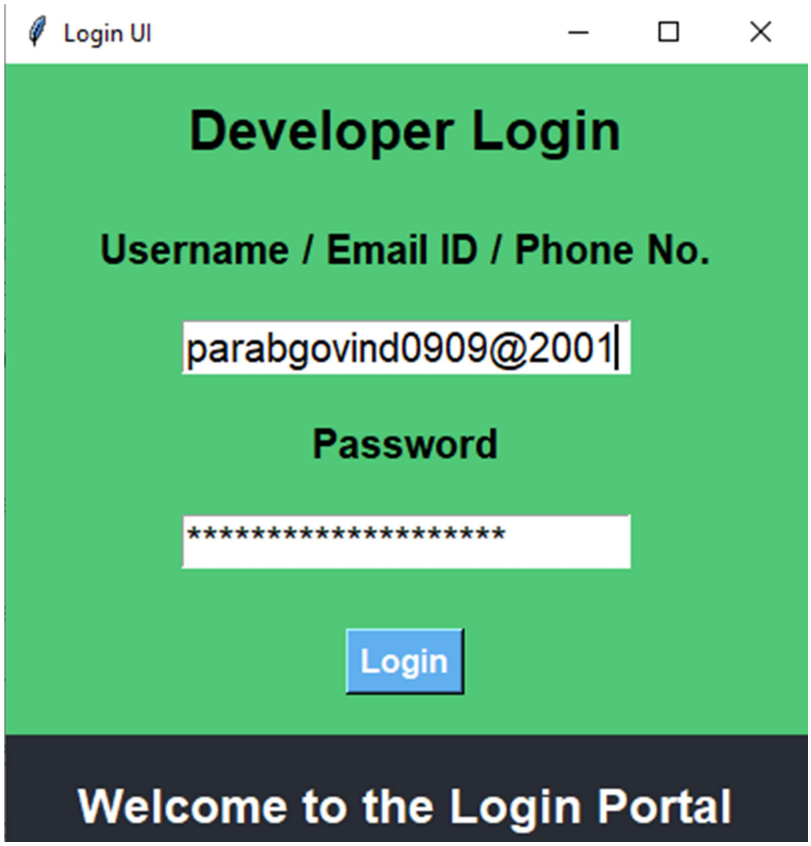
## Result

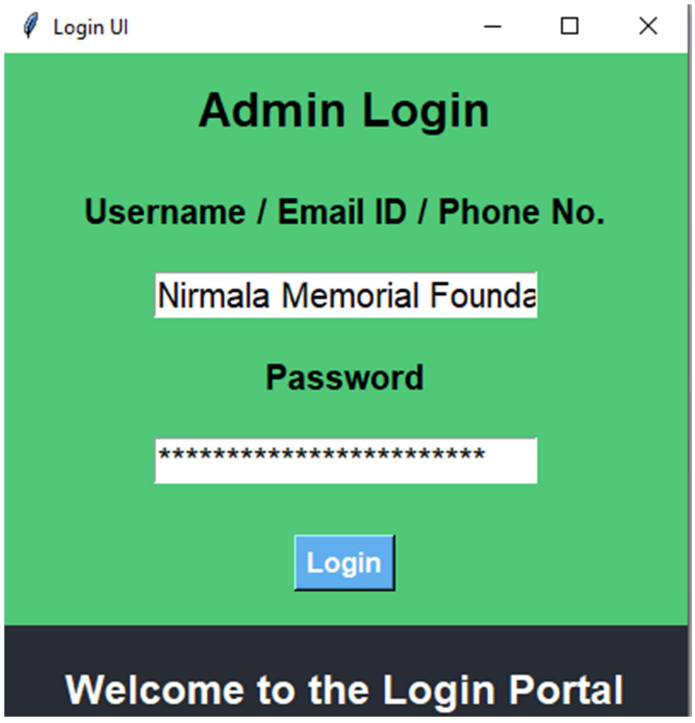
**Login Section:**

In this section, clicking the "Developer" button prompts you to input the developer's login credentials. If they match those stored in the system, you'll be directed to the developer section. Clicking the "Admin" button similarly requires entering the admin's login credentials. Upon a match with the stored credentials, you'll be directed to the admin section. Clicking the "Student" button directly navigates you to the student section.

If you enter wrong credentials then error message will be displayed.

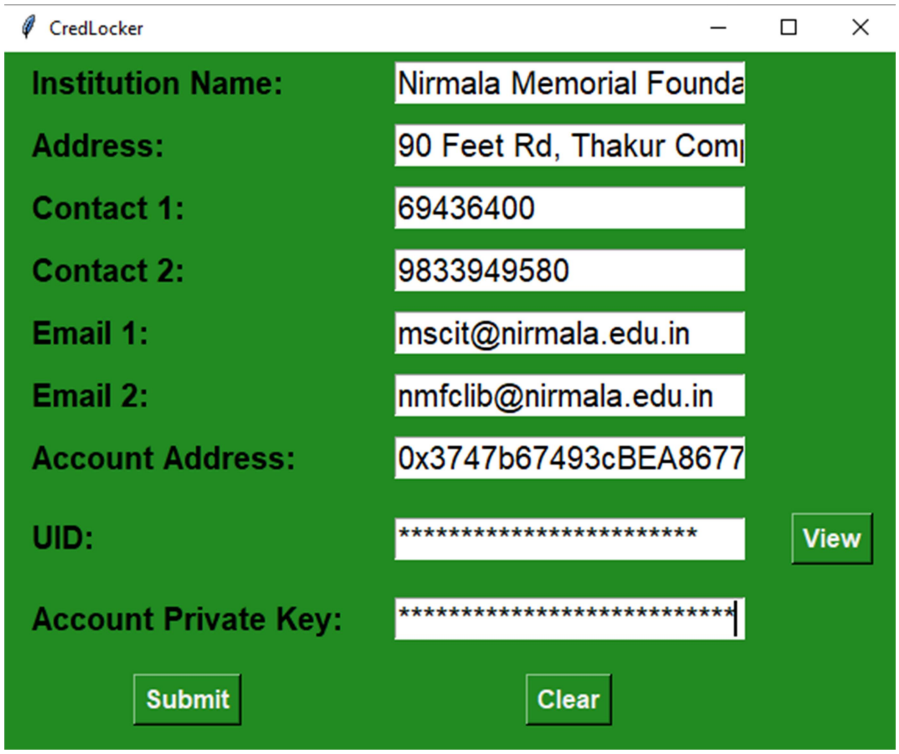


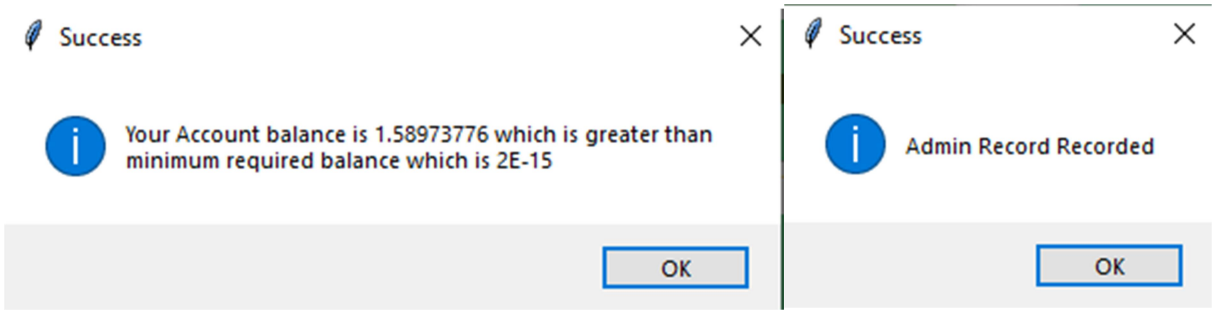

Login UI
Developer Login
Username / Email ID / Phone No.
parabgovind0909@2001
Password
**********
Login
Welcome to the Login Portal


Login UI
Admin Login
Username / Email ID / Phone No.
Nirmala Memorial Founda
Password
**********
Login
Welcome to the Login Portal

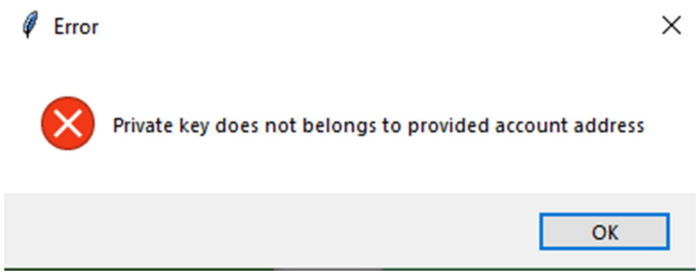
**Developer Section:**

Access to this page is granted only upon successful login as a developer. On this page, developers, or maintainers, can communicate with various entities and add legitimate organizational data to register new entities. This will assist other entities seeking to verify the credentials. In this section to add data the Ethereum wallet balance and address will be checked to make sure that wallet data is valid and has minimum required balance.

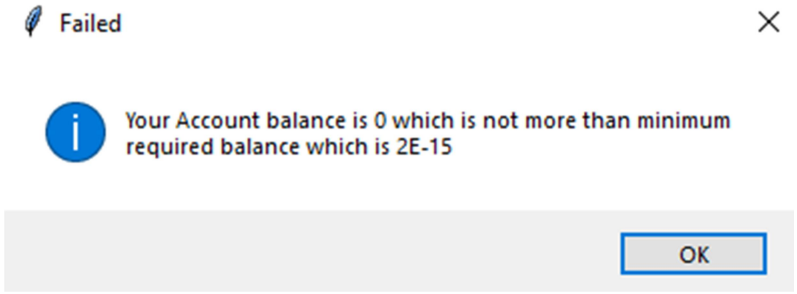



If address or private key data is invalid then error message will be displayed.



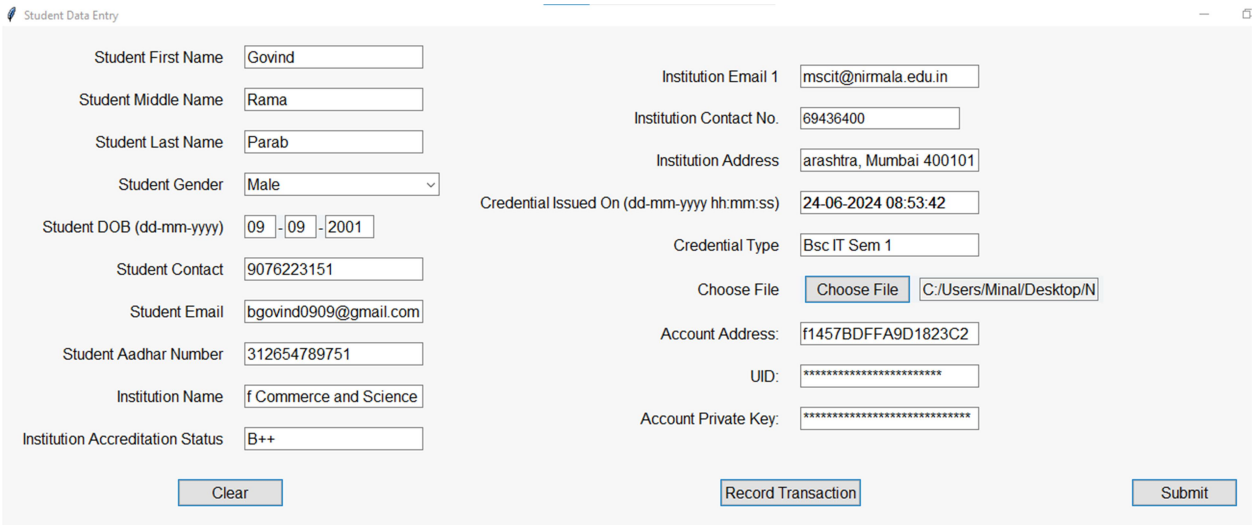

If address or private key data is valid but does not have sufficient balance then also error message will be displayed.

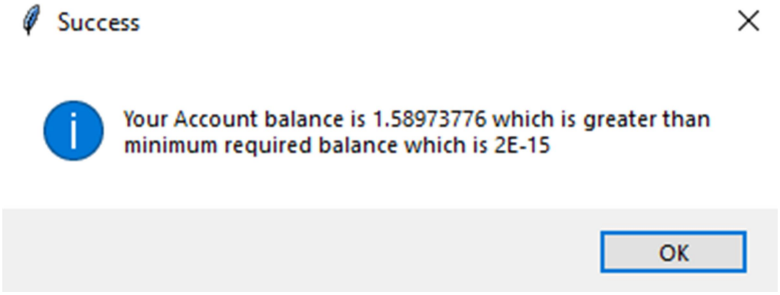


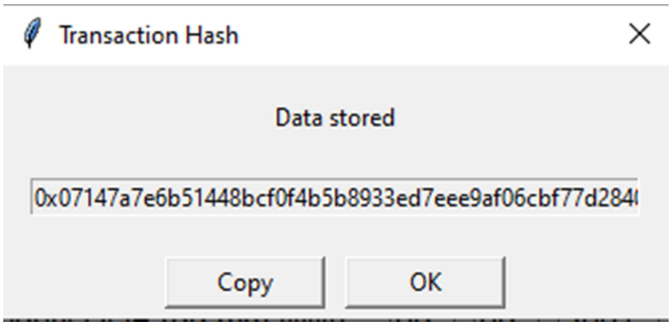

**Admin Section:**

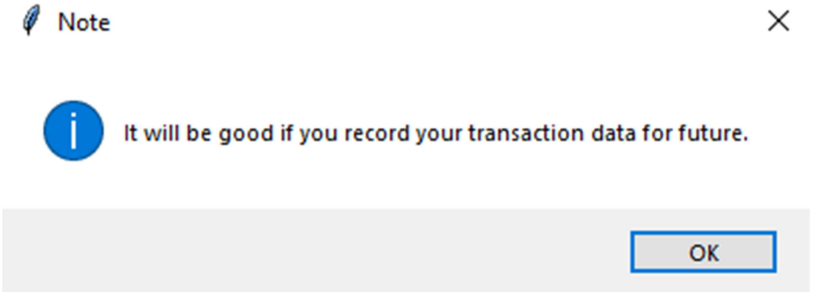
Access to this page is granted only upon successful login as an admin. On this page, admins or institutions can add legitimate organizational and student credential data to the system. This data will assist other entities in verifying credentials. In this section, to add data, the Ethereum wallet balance and address will be checked to ensure validity and that it meets the minimum required balance. A certain amount of Ether will be charged for this process. Also upon successful addition of data the unique hash id will be provided which is essential to retrieve or verify the stored data in future.

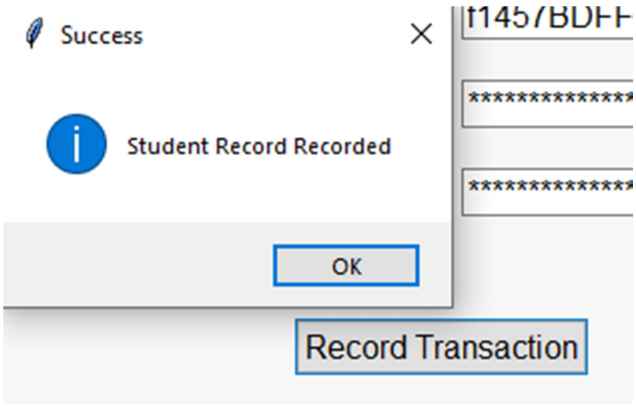
This section also allows users to store the data in distributed database so that if student lost their hash id of document they will be still able to retrieve it from blockchain.


Student Data Entry
Student First Name
Govind
Student Middle Name
Rama
Student Last Name
Parab
Student Gender
Male
Student DOB (dd-mm-yyyy)
09 - 09 - 2001
Student Contact
9076223151
Student Email
bgovind0909@gmail.com
Student Aadhar Number
312654789751
Institution Name
f Commerce and Science
Institution Accreditation Status
B++
Institution Email 1
mscit@nirmala.edu.in
Institution Contact No.
69436400
Institution Address
arashtra, Mumbai 400101
Credential Issued On (dd-mm-yyyy hh:mm:ss)
24-06-2024 08:53:42
Credential Type
Bsc IT Sem 1
Choose File
Choose File
C:/Users/Minal/Desktop/N
Account Address:
f1457BDFFA9D1823C2
UID:
Account Private Key:
Clear
Record Transaction
Submit


Success
Your Account balance is 1.58973776 which is greater than minimum required balance which is 2E-15
OK


Transaction Hash
Data stored
0x07147a7e6b51448bcf0f4b5b8933ed7eee9af06cbf77d284
Copy
OK


Note
It will be good if you record your transaction data for future.
OK

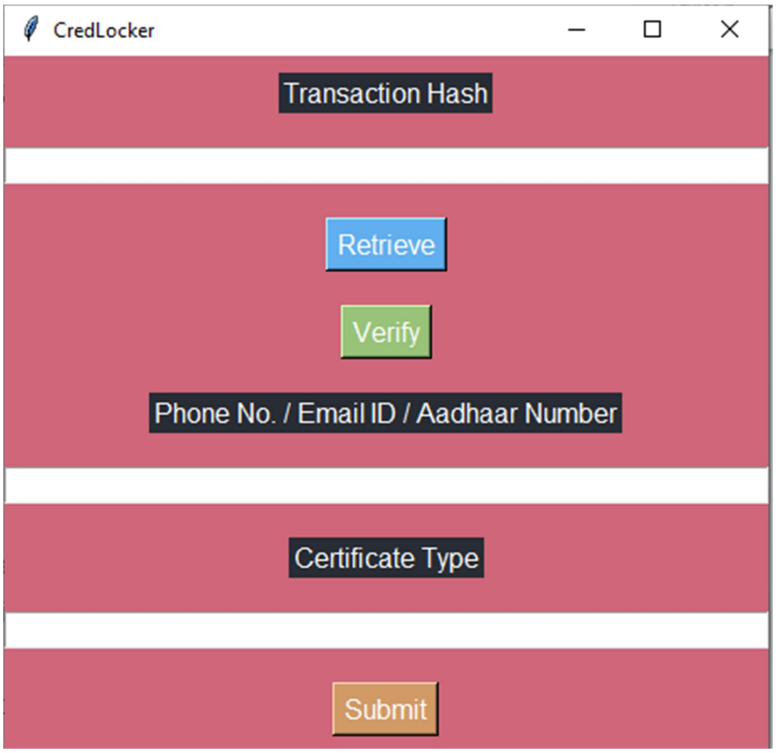



For each input entry basic validation is implemented. Account balance and private is checked and a very little amount of fee is charged.

If address or private key data is invalid then error message will be displayed.

If address or private key data is valid but does not have sufficient balance then also error message will be displayed.

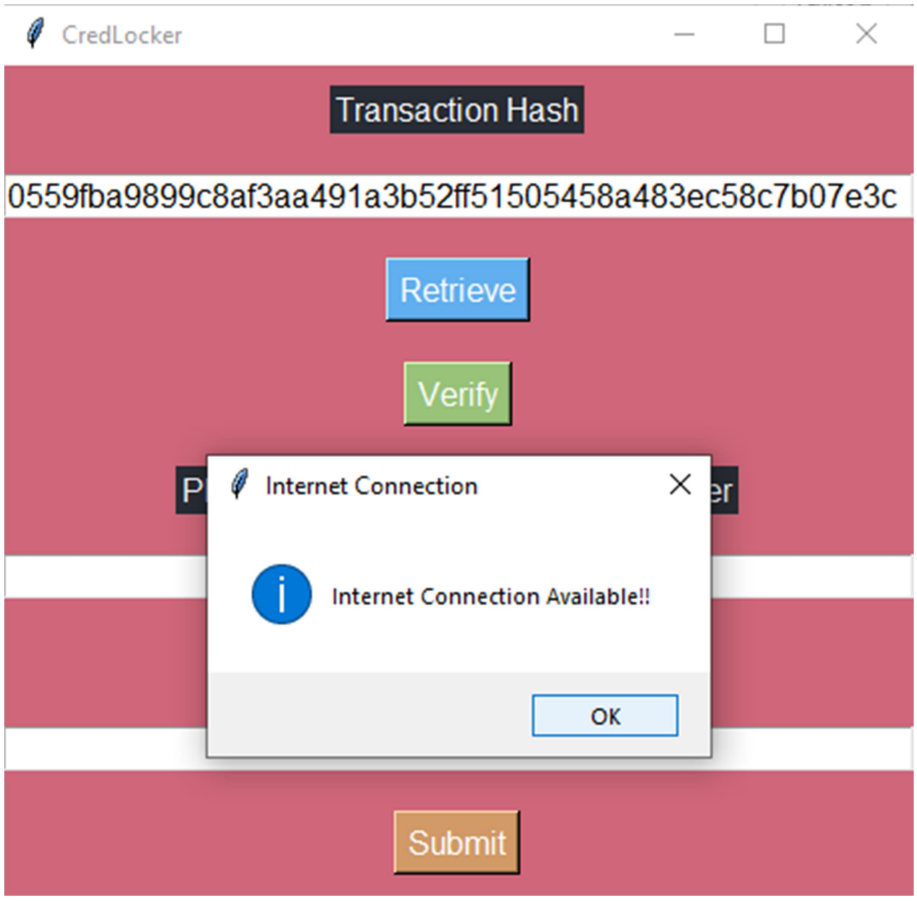
Student Section:

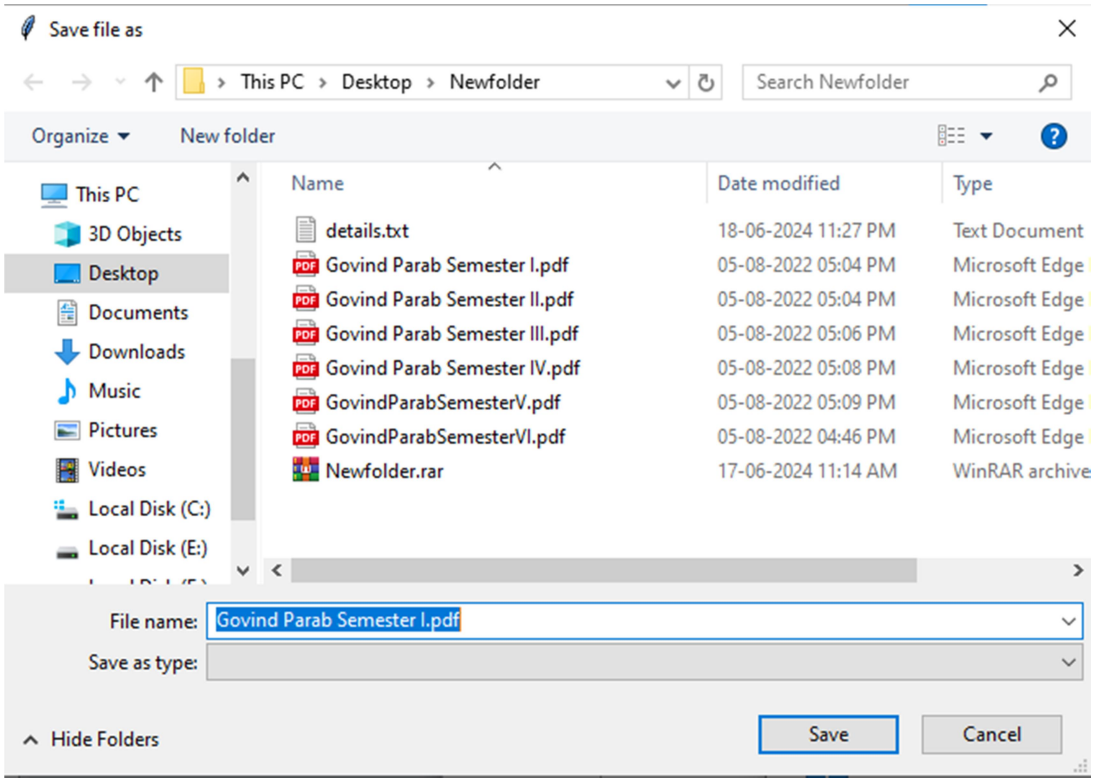



This section consists three major parts:

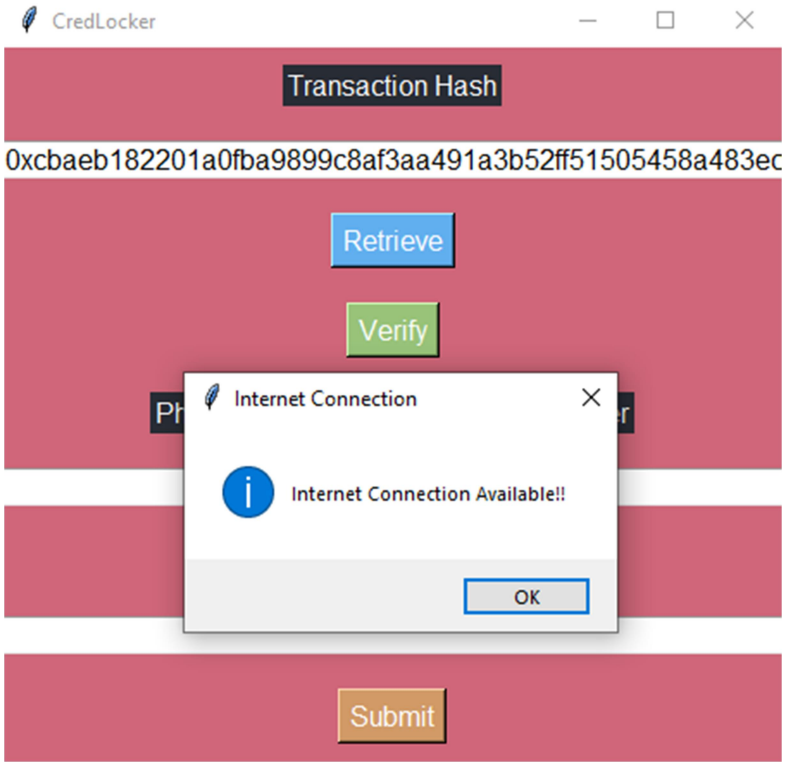
The first part is credential retrieval part were the students or other entities hash to put the hash associated with a particular certificate or any other document in transaction hash input field and then click on retrieval button.

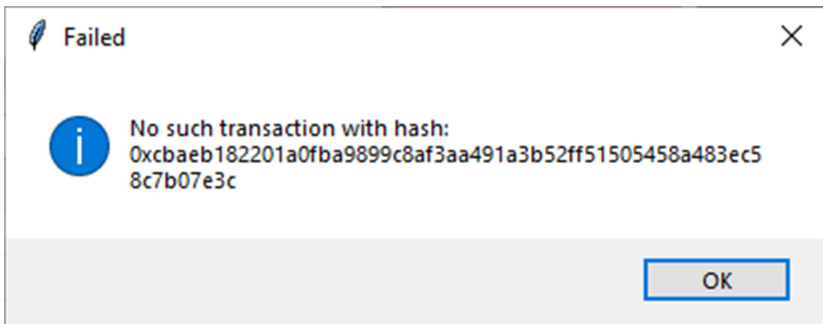
If its valid hash then the save file dialog box will appear and you can save that credential to your local device as shown below:





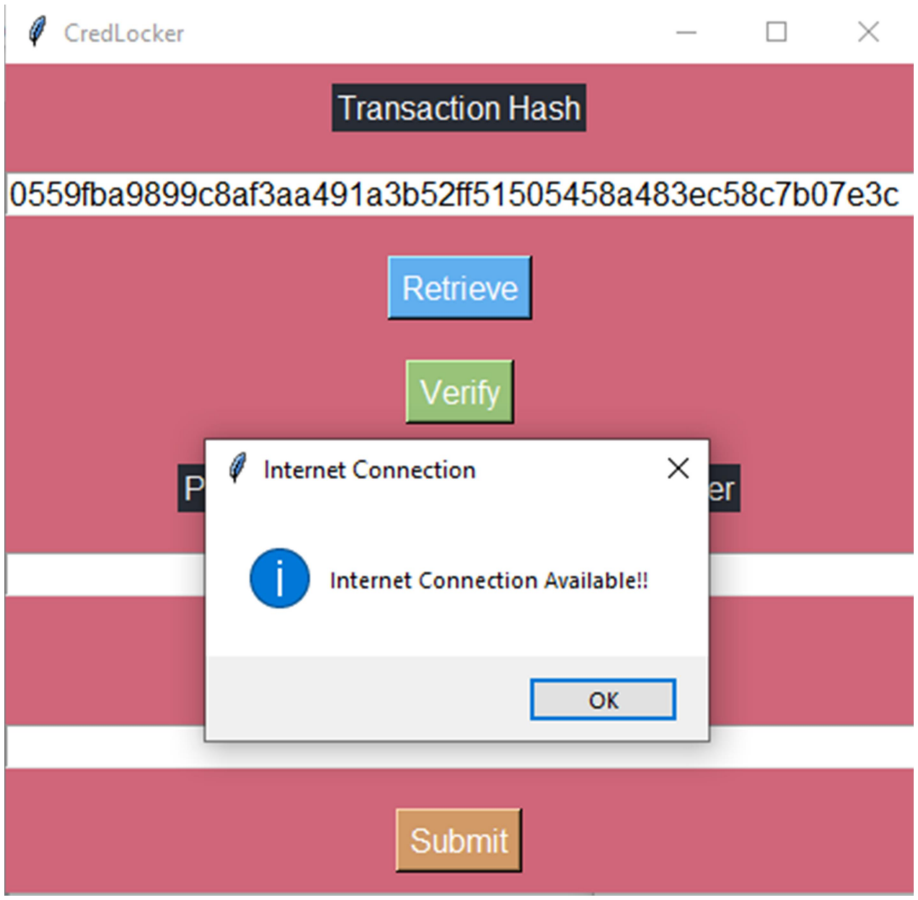

If the hash isn't valid then the error message will be displayed as shown below:

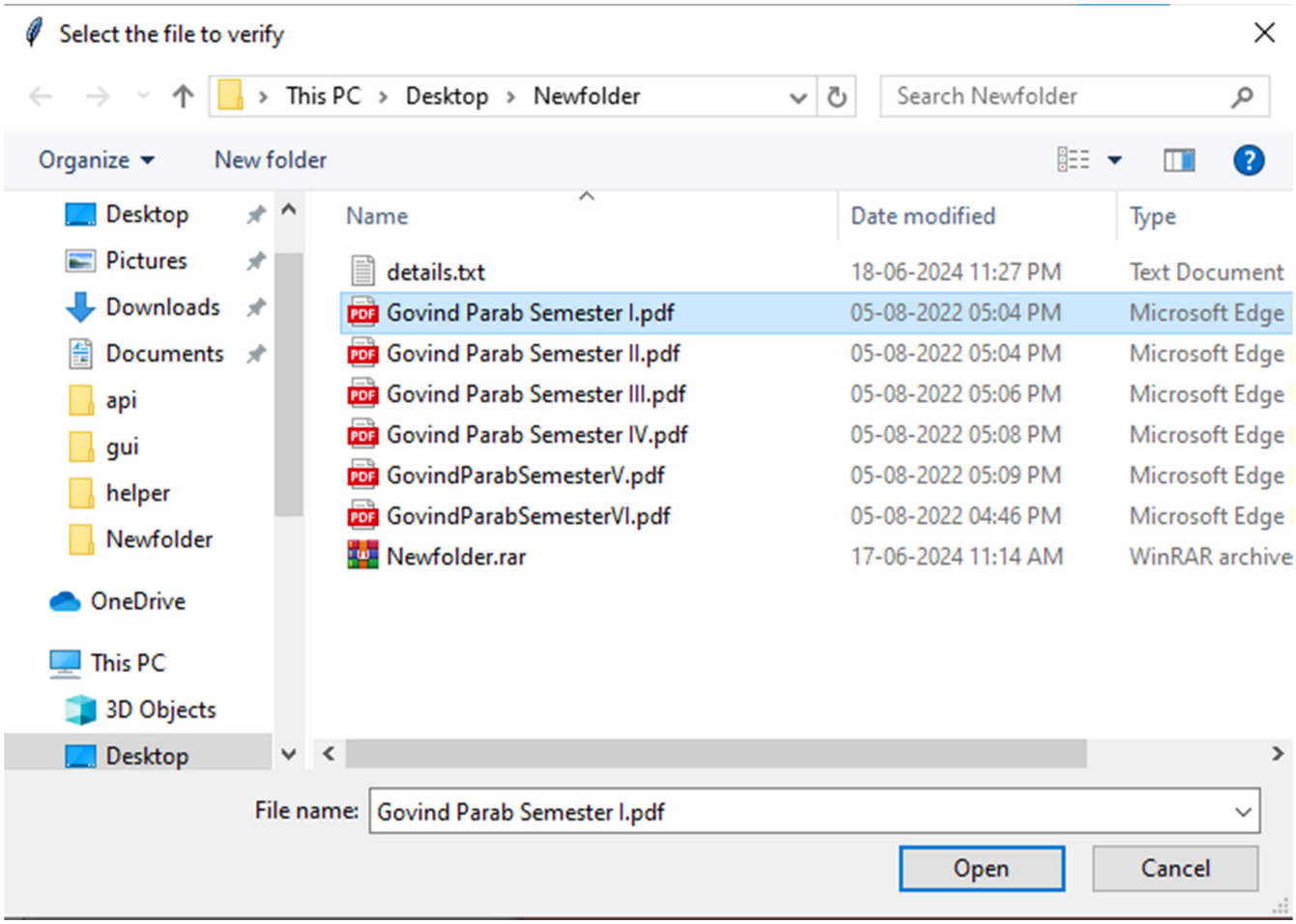


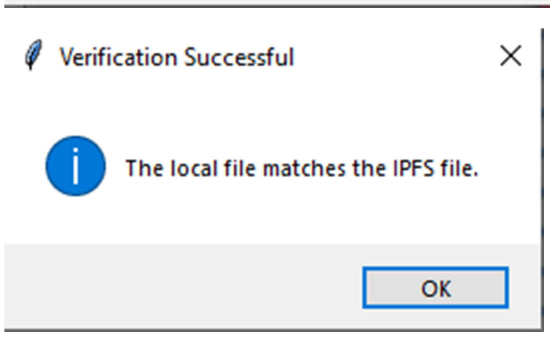



The second part involves using the "verify" button. Here, students or other entities can verify their local copy of a certificate against the one stored on the blockchain. To do this, they input the hash into the transaction hash field and click on the verify button.

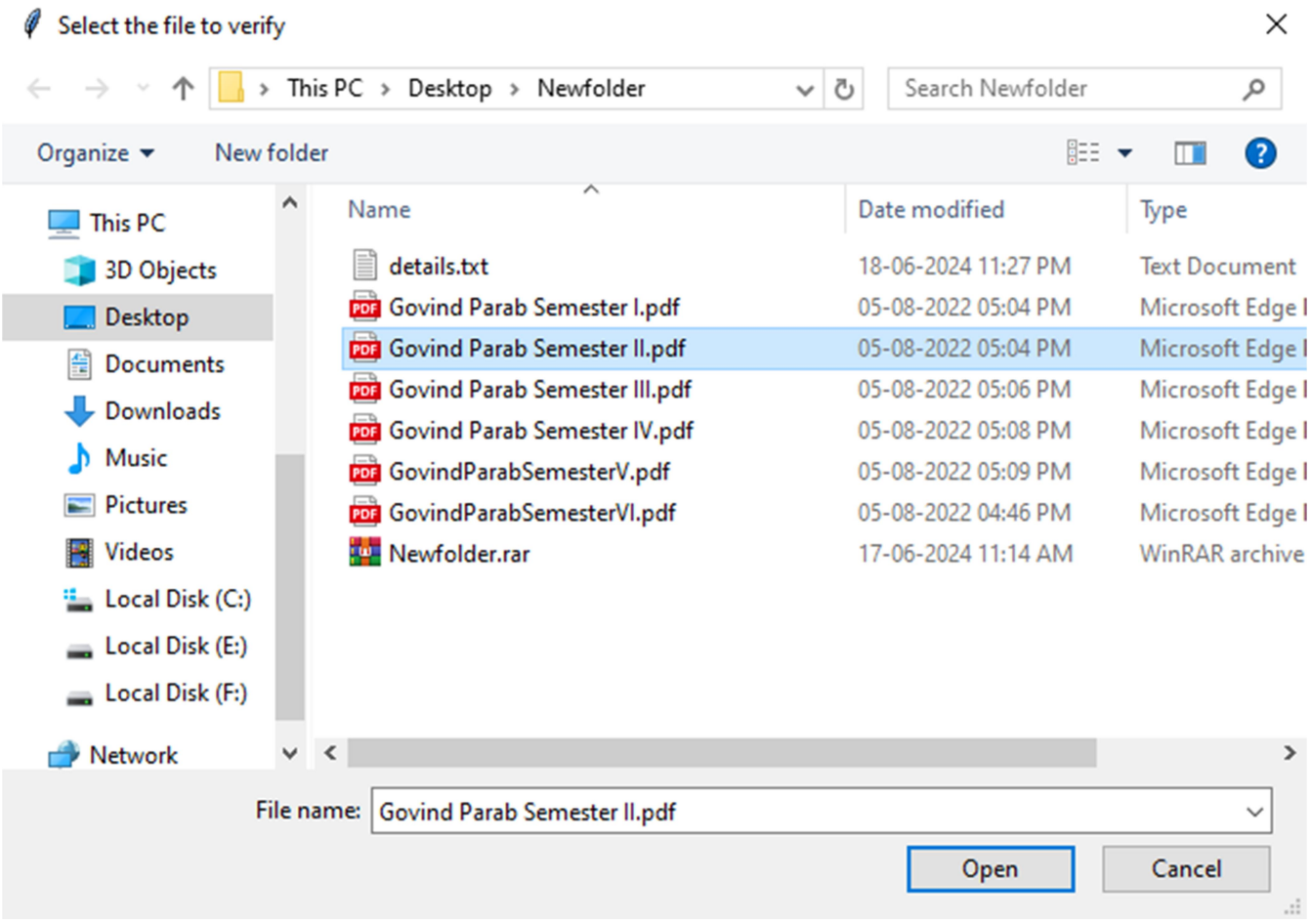
If the hash is valid, a dialog box will appear prompting the user to select the local file for comparison with the securely stored file. If the contents match, a success message will be displayed.

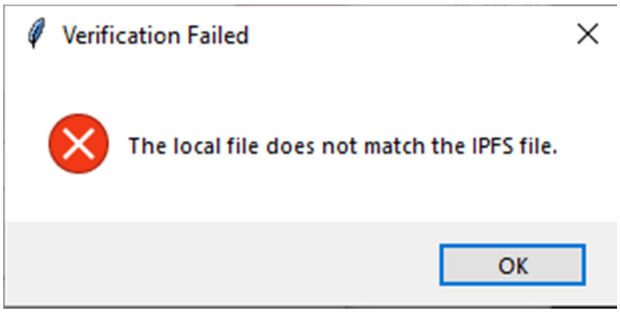




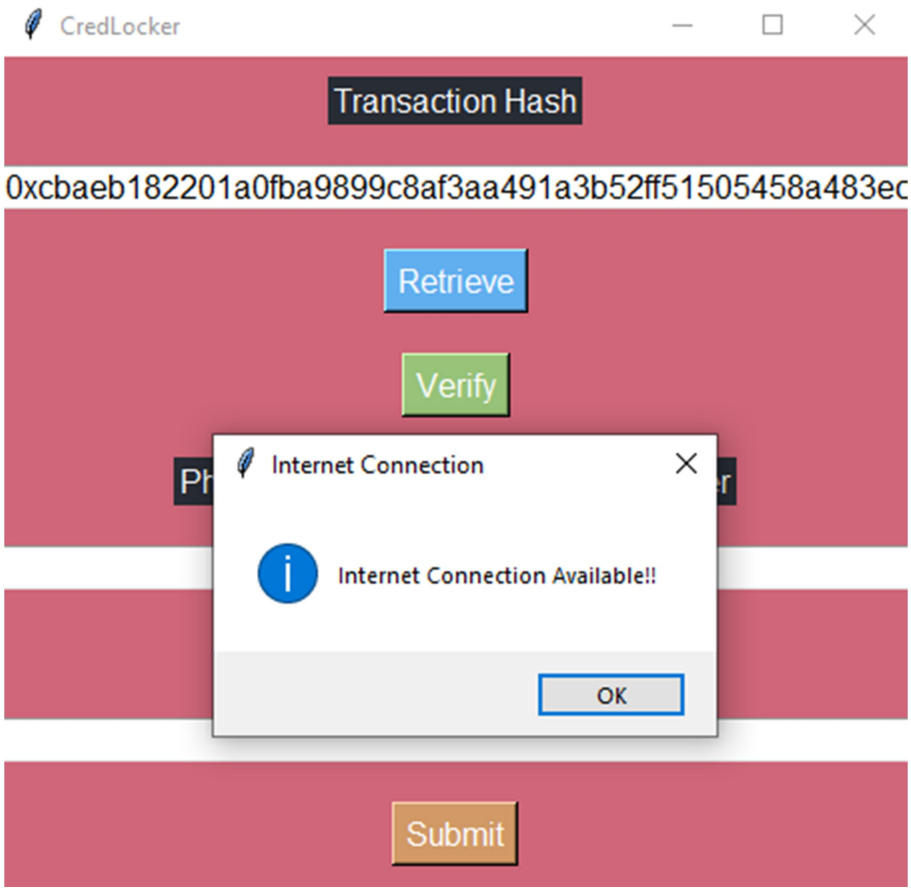



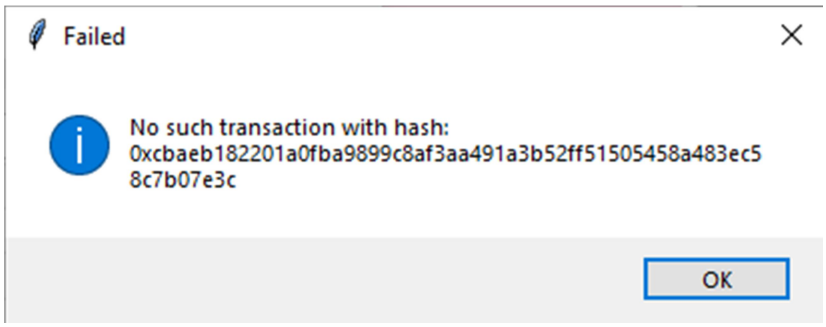

If the content doesn’t match then it will show a failure message.



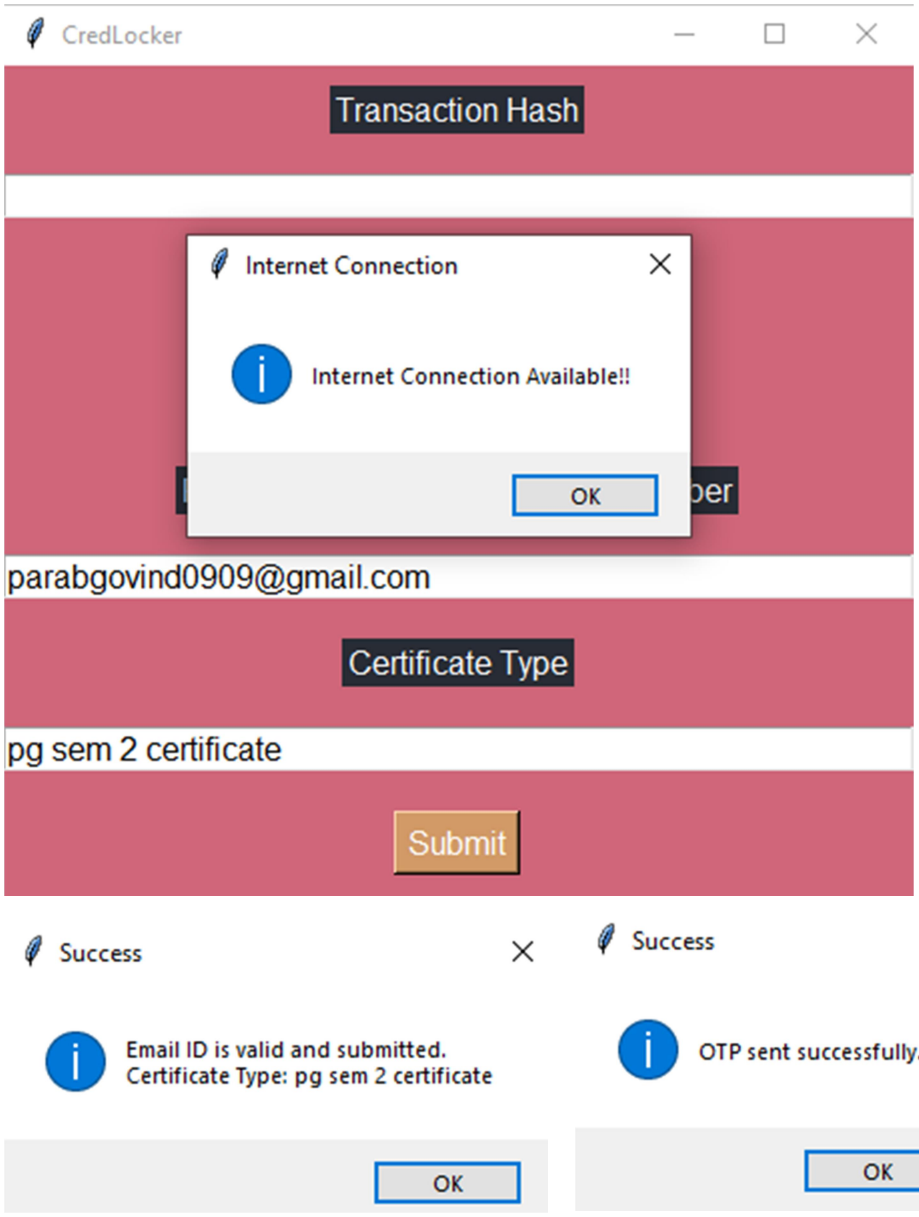




If the hash is invalid, an "invalid hash" message will be displayed.



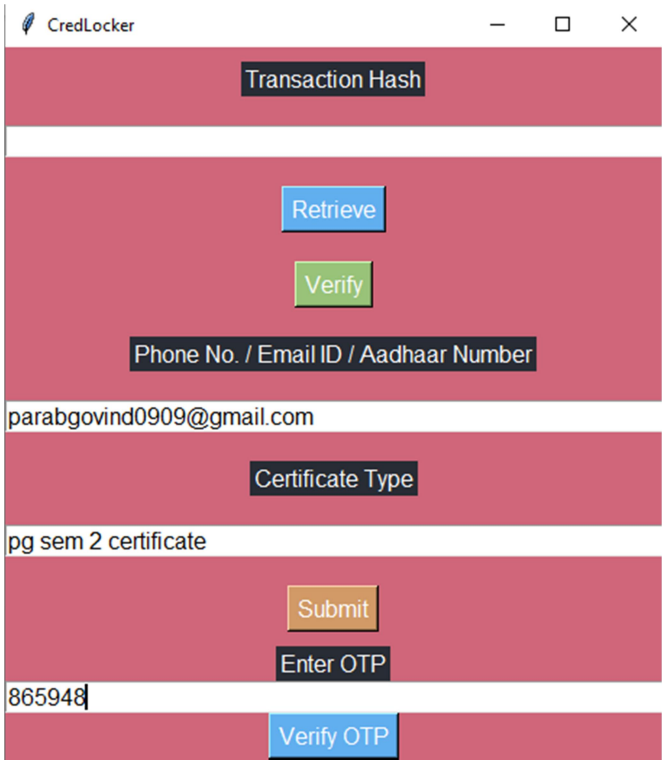



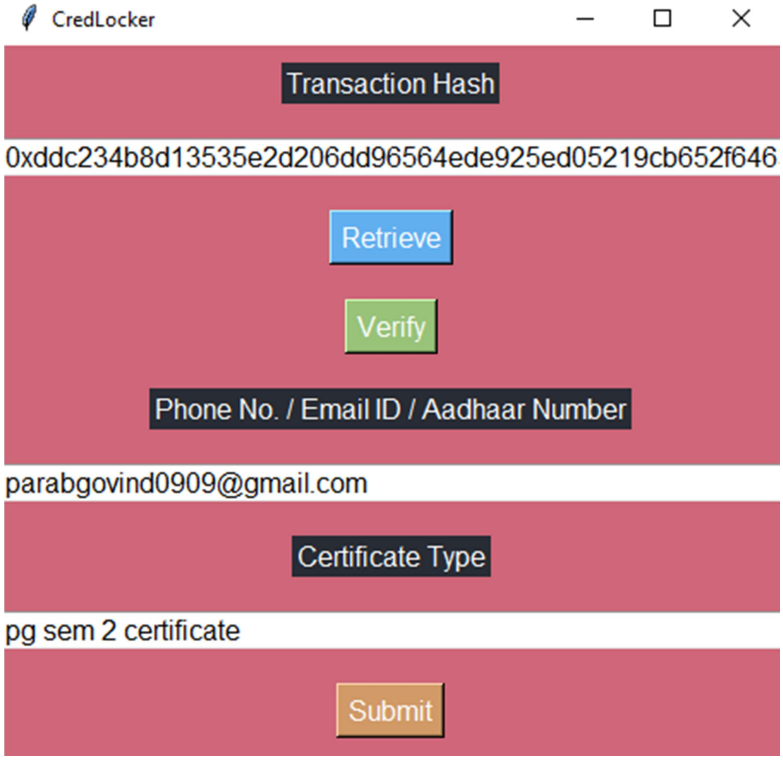

The Third part is ask the students to input their email-id, contact no. or aadhaar no. (for now only email-id is accepted) and exact name of the certificate type which if is stored on blockchain then the OTP for user verification will be sent on email and if its valid then the transaction hash will be provided in the transaction field further allowing them to do retrieval or verify operation.


CredLocker
Transaction Hash
Internet Connection
Internet Connection Available!!
OK
parabgovind0909@gmail.com
Certificate Type
pg sem 2 certificate
Submit

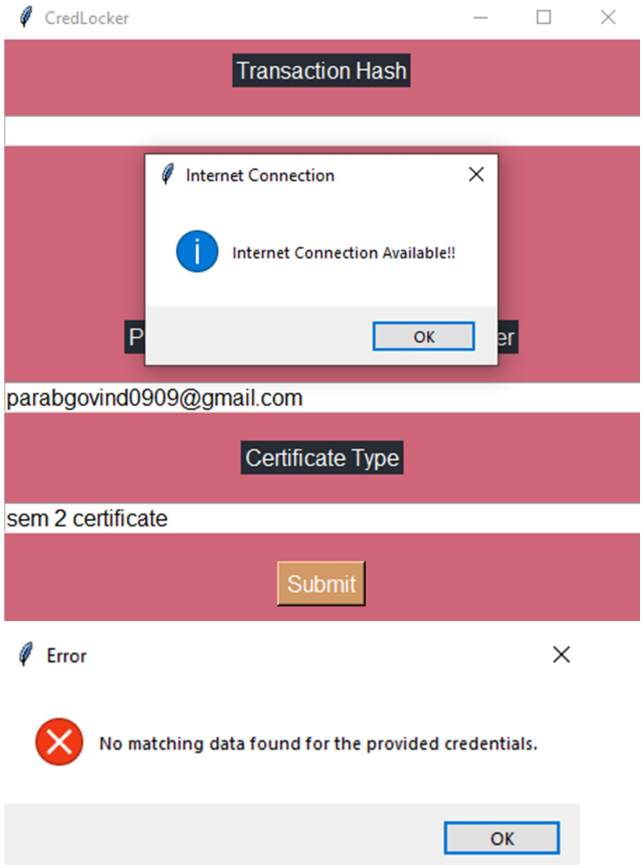

Success
Email ID is valid and submitted.
Certificate Type: pg sem 2 certificate
OK


Success
OTP sent successfully.
OK

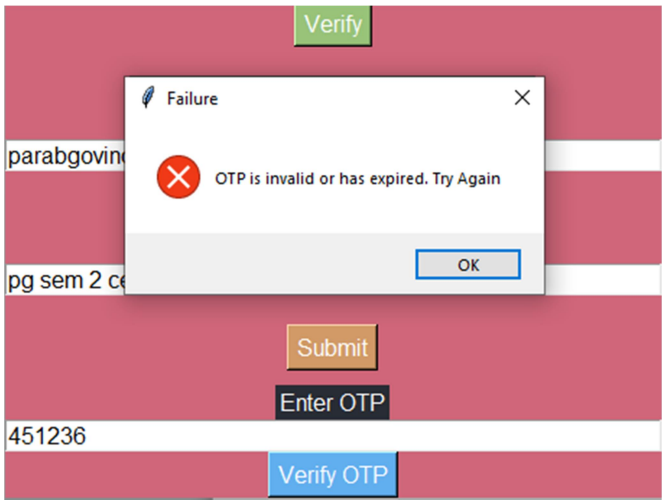



If it is not valid then the error message will be displayed.




If OTP is invalid then also an error message will be displayed.




This is particularly focus the need for another retrieval mechanism if the students lost their hash id of document.

## Conclusion

In conclusion, the "Enhancing Academic Credentials Security and Verification Using Blockchain" project marks a ground-breaking effort in utilizing blockchain technology, particularly the Ethereum Sepolia Testnet, to strengthen the security, authenticity, and management of academic certificates. This initiative addresses the urgent need to combat forgery and tampering in credentialing processes, thereby fostering greater trust and transparency among educational institutions, employers, and other stakeholders. The project's primary objectives were to transform academic certificates into tamper-proof digital signatures and to establish a blockchain-based infrastructure ensuring the immutability and verifiability of credentials. By achieving these goals, the project addresses longstanding vulnerabilities in traditional credentialing systems and sets new standards for reliability and efficiency in verifying academic qualifications.

The project's comprehensive scope included meticulous design, development using Solidity for smart contracts, integration with existing platforms, rigorous testing, thorough documentation, and comprehensive training of stakeholders. The choice of the Ethereum Sepolia Testnet provided a robust simulated environment for testing and validating smart contracts and decentralized applications before mainnet deployment. Essential tools such as Firebase Realtime Database, Infura, and Pinata API highlighted the project's commitment to leveraging advanced technologies for optimal performance and reliability. The applicability of blockchain technology in this context extends beyond academic institutions to include employers, professional certification bodies, and individuals, enhancing operational efficiency and mitigating risks associated with credential fraud and data breaches. The process involves universities adding students' certificates to the blockchain, creating a secure and immutable record of academic achievements. Verification is facilitated using primary identifiers such as the student's aadhaar number, email-id, contact number, or transaction ID, ensuring that only legitimate parties can access and verify the credentials.

Looking ahead, the success of this project serves as a compelling case study for future advancements in credential management and verification systems. Potential enhancements include integrating advanced cryptographic techniques to further secure data, developing user-

friendly interfaces to simplify the verification process for non-technical users, and expanding the system's interoperability with international educational institutions and credentialing bodies. Additionally, incorporating machine learning algorithms could automate the detection of fraudulent activities and streamline verification processes. Exploring partnerships with industry leaders and government bodies could also facilitate widespread adoption and standardization of blockchain-based credential verification. As blockchain technology continues to evolve, these enhancements will contribute to the on-going improvement of credentialing systems, ensuring that academic credentials remain secure, reliable, and globally recognized.

## References

1. Rustemi, Avni, Fisnik Dalipi, Vladimir Atanasovski, and Aleksandar Risteski. "A Systematic Literature Review on Blockchain-Based Systems for Academic Certificate Verification." IEEE Access (2023).

2. Kaneriya, Jayana, and Hiren Patel. "A Secure and Privacy-Preserving Student Credential Verification System Using Blockchain Technology." International Journal of Information and Education Technology 13, no. 8 (2023).

3. Tariq, Aamna & Binte Haq, Hina & Ali, Syed. (2019). Cerberus: A Blockchain-Based Accreditation and Degree Verification System.

4. Leka, Elva & Selimi, Besnik. (2020). BCERT - A Decentr kialized Academic Certificate System Distribution Using Blockchain Technology. 12. 103-118.

5. Ayub Khan, Abdullah, Asif Ali Laghari, Aftab Ahmed Shaikh, Sami Bourouis, Amir Madany Mamlouk, and Hammam Alshazly. 2021. "Educational Blockchain: A Secure Degree Attestation and Verification Traceability Architecture for Higher Education Commission" Applied Sciences 11, no. 22: 10917. https://doi.org/10.3390/app112210917

6. Kistaubayev, Yerlan, Galimkair Mutanov, Madina Mansurova, Zhanna Saxenbayeva, and Yassynzhan Shakan. 2023. "Ethereum-Based Information System for Digital Higher Education Registry and Verification of Student Achievement Documents" Future Internet 15, no. 1: 3. https://doi.org/10.3390/fi15010003

7. Rahman, Tasfia, Sumaiya Islam Mouno, Arunangshu Mojumder Raatul, Abul Kalam Al Azad, and Nafees Mansoor. "Verifi-chain: A credentials verifier using blockchain and IPFS." In International Conference on Information, Communication and Computing Technology, pp. 361-371. Singapore: Springer Nature Singapore, 2023.

8. Raghavender, K. V., S. Alankruthi, A. Akhila, T. Preethi, and M. Ashritha. "Decentralized Smart Contract Certificate System Using Ethereum Blockchain Technology." In Second International Conference on Emerging Trends in Engineering (ICETE 2023), pp. 452-461. Atlantis Press, 2023.

9. Sultana, Shaik Arshiya, Chiramdasu Rupa, Ramanadham Pavana Malleswari, and Thippa Reddy Gadekallu. "Ipfs-blockchain smart contracts based conceptual framework to reduce certificate frauds in the academic field." Information 14, no. 8 (2023): 446.