A REPORT
ON

"Design of a Privacy-preserving Mobile Wallet for Self-Sovereign Identity Management"

Submitted by,

Ms. DYAMALLI K - 20211CBC0046

Under the guidance of,

Ms. ARSHIYA LUBNA

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (BLOCK CHAIN)

At

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship report "**Design of a Privacy-preserving Mobile Wallet for Self-Sovereign Identity Management**" being submitted by "DYAMALLI K" bearing roll number "20211CBC0046" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering(Block Chain) is a bonafide work carried out under my supervision.

Ms. ARSHIYA LUBNA
Assistant Professor
PSCS
Presidency University

Dr. S. PRAVINTH RAJA
Professor & HoD
PSCS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor - Engineering
Dean –PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled **"Design of a Privacy-preserving Mobile Wallet for Self-Sovereign Identity Management" in** partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering (Block Chain)**, is a record of my own investigations carried under the guidance of **Ms. ARSHIYA LUBNA, ASSISTANT PROFESSOR, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	**Roll No**	**Signature of the Student**
Dyamalli K	20211CBC0046	

ABSTRACT

Concern over protecting user autonomy, privacy, and security is growing as digital identification systems continue to develop. By introducing a mobile wallet that protects privacy for Self-Sovereign Identity (SSI) management, this project enables people to safely store, maintain, and distribute their digital credentials without relying on centralized authority. The wallet allows users to create self-owned, verifiable identities while maintaining data security and integrity by utilizing Verifiable Credentials (VCs) and Decentralized Identifiers (DIDs).

The wallet uses Selective Disclosure, which allows users to share only particular identification information rather than whole credentials, to improve privacy. Zero-Knowledge Proofs (ZKPs) also enable authentication without disclosing private information. End-to-end encryption, secure key management, and multi-factor authentication (MFA) are further security measures that guard against unwanted access. The system is supported by blockchain technology, which guarantees immutability, transparency, and confidence in the issuing and verification of credentials.

Identity verification, online authentication, financial transactions, and access control are just a few of the uses for this mobile wallet, which was made with interoperability, usability, and privacy protection in mind. The system offers a user-centric approach to digital identity management by using decentralized identity concepts, which also improve security, build trust, and comply with contemporary data protection laws.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and "**Dr. S. PRAVINTH RAJA**" Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Ms. Arshiya Lubna, Assistant Professor,** and Reviewer **Ms. Ashishika Singh, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 University Internship Coordinators **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Internship Coordinator **Ms. Ashishika Singh** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

DYAMALLI. K

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The importance of maintaining security and privacy has increased with the usage of digital identity systems. Traditional identity management frameworks are susceptible to identity theft, data breaches, and unlawful eavesdropping since they usually depend on centralized authorities. Additionally, customers are often asked to provide more personal information than is necessary, which raises concerns about privacy violation. Self-Sovereign Identity (SSI) has been developed to solve these issues by granting users total control over their online identities without the involvement of other parties.

A privacy-focused mobile wallet is a crucial component of SSI, providing users with a secure means of managing, sharing, and storing their login information. Unlike traditional digital identity methods, this mobile wallet's decentralized structure allows users to verify their credentials without revealing unnecessary personal information. Such a solution is particularly beneficial in industries like finance, healthcare, governance, and digital services where secure and confidential identity verification is essential.

1.1 The necessity of a mobile wallet that protects privacy :

Not with standing SSI's benefits, its actual implementation requires an efficient and discreet identity wallet. A well-designed SSI-based mobile wallet should have the following features: Decentralization entails reducing dependence on centralized organizations in order to increase security and reduce the danger of single points of failure.
User autonomy is the ability for individuals to manage their credentials independently. Selective disclosure is the practice of letting users reveal only the most important identifying characteristics rather than all of their personal data.
Strong Security Measures: Cryptographic techniques including encryption, blockchain-based authentication, and Zero-Knowledge Proofs (ZKPs) are employed to ensure data integrity and confidentiality. Interoperability is the ability to support several identity verification methods for seamless cross-platform use.

1.2 The purpose of the study:

The project's objective is to provide a mobile wallet that enhances user control, security, and interoperability while safeguarding privacy. Among the primary objectives are: establishing a secure mobile wallet that enables decentralized management and storage of digital credentials. implementing selective disclosure techniques to allow users to reveal only the information that is required for identification. enhancing identity verification through the use of blockchain technology. enhancing security and trust by using cryptographic techniques to protect user credentials.

1.3 Study Area:

To provide secure identity verification and authentication, the proposed mobile wallet will leverage blockchain technology and privacy-enhancing cryptographic approaches. Users will be able to: Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) should be handled and kept securely. Provide as little personal information as you can while confirming your identity with service providers. Interact with verifiers and issuers in a decentralized, untrustworthy ecosystem. Protect yourself from identity theft and unauthorized data access. By developing a secure, decentralized, and privacy-enhancing identification solution, this study contributes to the advancement of trustworthy digital identity frameworks that prioritize user control and data protection. The study's findings will aid in the creation of digital identity systems that preserve privacy while meeting evolving legal and technological standards.

Chapter 2

LITERATURE SURVEY

2.1 Identity Self-Sovereignty (SSI):

A decentralized method of identity management called Self-Sovereign Identity (SSI) gives people complete authority over their online personas without the need for centralized oversight. By allowing users to control their Verifiable Credentials (VCs) and Decentralized Identifiers (DIDs), it guarantees safe and authentic identity transactions.

The adoption of SSI is supported by a number of frameworks, such as the W3C Verifiable Credentials (VC) and Decentralized Identifiers (DID) standards, which set internationally recognized standards for digital identification. While uPort is an Ethereum-based identification solution that lets users establish and manage their credentials on their own, Hyperledger Indy offers a blockchain-based infrastructure for decentralized identity. These technologies aid in the creation of an identity management system that is transparent, reliable, and user-focused. [1]

2.2 Identity Management Trust Chain:

By guaranteeing safe and verifiable identification transactions across decentralized networks, a trust chain fortifies SSI. It improves authentication without centralized management by creating a tamper-resistant chain of identity proofs through cryptographic linkages.

A permissioned blockchain network called Hyperledger Fabric uses smart contracts to verify credentials and protect privacy, enabling decentralized identity verification. In a similar vein, trust-based authentication models improve identity security by utilizing trust ratings and cryptographic attestations. According to research, combining decentralized identity solutions with trust-based models enhances authentication procedures, reduces fraud, and creates a trustworthy identity verification system. [2]

2.3 Techniques for Preserving Privacy:

Identity management requires privacy protection, and a number of cryptographic techniques support safe yet private authentication:

One cryptographic method that allows users to demonstrate credentials ownership without disclosing real information is called Zero-Knowledge Proofs, or ZKPs.

This can be used in situations such as age verification, when an individual can verify their eligibility without disclosing their exact birthdate.

• Selective Disclosure: Reduces needless data exposure by enabling people to divulge only the essential identifying information rather than complete credentials.

• Commitment Schemes: These offer privacy-preserving cryptographic guarantees that identification attributes don't alter.

While adhering to decentralized identification standards, these privacy-enhancing technologies aid in security maintenance. [3]

2.4 Security and Mobile Wallets:

Because they enable users to safely store and exchange credentials, mobile wallets are essential for managing decentralized identities. To guarantee identity protection, these wallets must include secure key management, offline verification, and robust encryption.

Although Evernym and uPort, two popular SSI-based mobile wallets, provide decentralized identification solutions, they sometimes lack strong privacy-focused features.

Features like multi-factor authentication (MFA), biometric authentication, and end-to-end encryption are crucial for improving security.

Furthermore, identity theft and unwanted access can be avoided with hardware-based security modules that provide secure key management. [4]

2.5 Difficulties and Prospects:

Despite significant progress in blockchain-based identity management and Self-Sovereign Identity (SSI) systems, several challenges persist. Scalability remains a concern, as decentralized verification methods can face difficulties handling large numbers of users and transactions efficiently, potentially causing slowdowns and higher costs.

Additionally, the diversity of blockchain platforms leads to interoperability issues, making it hard for different systems to communicate and work together smoothly.

Another challenge lies in balancing the decentralized nature of these systems with compliance to data protection laws such as GDPR. These regulations impose requirements on how personal data is stored, accessed, and deleted, which can conflict with the permanent and transparent characteristics of blockchain technology.

Addressing these challenges is crucial for advancing decentralized identity solutions while ensuring privacy and regulatory adherence.[5]

Chapter 3

RESEARCH GAPS OF EXISTING METHODS

Although there has been progress in digital identity management, many of the current approaches still struggle to ensure user autonomy, privacy, security, and decentralization. Traditional and decentralized identity solutions aim to increase security, but their practical implementation usually falls short. The primary research gaps in the identity management systems now in use are listed below:

3.1 Dependency on centralized authority :

Most digital identity systems are reliant on centralized entities, such as corporations, governmental bodies, or third-party identity providers. Because of the single points of failure that this dependency creates, these systems are susceptible to cyberattacks, data breaches, and illegal data sharing. Even with federated identity models, where a single entity manages authentication across several services, users still need to trust other providers with their sensitive data.

3.2 Inadequate Data Exchange User Independence :

In traditional identity verification processes, users often have limited control over their personal data. Because many systems request more information than is necessary, users are vulnerable to risks like identity theft and surveillance.

One major problem with present models is the absence of selective disclosure processes, which would allow users to reveal only the most important identifying characteristics needed for verification.

3.3 Inadequate Systems for Protecting Privacy :

Existing identity systems sometimes require complete credential disclosure during verification, making it difficult to preserve user privacy.

Users are exposed to needless data because cryptographic techniques like Selective Disclosure and Zero-Knowledge Proofs are not widely used.

3.4 Insufficient Interoperability of Identity Systems :

It might be difficult to verify credentials submitted on one platform on another since many identity management systems operate in different ecosystems. In a number of sectors, such as e-governance, healthcare, and finance, simple identity verification is hampered by the lack of cross-platform compatibility and standards.

It is still challenging to attain widespread adoption without embracing global standards such as Decentralized Identifiers and Verifiable Credentials.

3.5 Vulnerabilities in Security and Authentication :

Since many systems still utilize password-based authentication, it is quite susceptible to attacks like phishing and credential leaks. Even if biometric authentication methods are more secure, there is still a chance that biometric data could be hacked.

Blockchain-based identity solutions that are poorly designed may also expose users to smart contract vulnerabilities and data correlation problems, jeopardizing transaction privacy.

3.6 High processing and storage costs:

Some decentralized identity systems require a lot of processing power and storage, especially those that employ blockchain for data verification. Storage of identifying credentials on-chain may lead to scalability issues, increased costs, and inefficiencies.

More efficient off-chain storage methods in conjunction with on-chain verification systems need to be researched in order to optimize performance without compromising security.

3.7 Ineffective Credential Revocation :

The lack of decentralized, real-time credential revocation mechanisms is a major flaw in many identity systems. When an identification certificate is lost, stolen, or needs to be updated, revocation may be slow, ineffectual, or non existent. A clear approach is required to ensure that compromised credentials can be revoked while maintaining system transparency and user confidence.

Chapter 4

PROPOSED MOTHODOLOGY

Using blockchain technology, cryptography, Verifiable Credentials (VCs), and Decentralized Identifiers (DIDs), the methodology focuses on developing a mobile wallet that safeguards privacy for managing Self-Sovereign Identity (SSI). This approach to identity management ensures decentralization, privacy, and security. The primary steps of the recommended methodology are as follows:

4.1 System Architecture and Design:

The mobile wallet will function as a decentralized identity management system and be composed of three primary parts:

Identity Issuers: Organizations such as banks, government offices, and educational institutions that give users Verifiable Credentials (VCs).

identity Holders (Users): People who selectively give their identification details when necessary and carefully retain their login credentials in their mobile wallet.

Verifiers: Businesses that validate credentials using blockchain-based methods to establish trust without relying on centralized databases.

The wallet will be built in compliance with the W3C DID and VC standards to enable cross-platform interoperability.

4.2 Using DIDs and VCs for Decentralized Identity Management DID Generation:

To ensure self-ownership of identification, users generate Decentralized Identifiers (DIDs) using cryptographic key pairs. Secure Verifiable Credential Storage: Off-chain, encrypted credentials from reliable sources are kept safe from unwanted access.

Selective disclosure that protects privacy allows users to reveal only the essential aspects of their identities rather than their whole login credentials.

4.3 Privacy-Preserving Authentication and Verification In order to improve security, the system will include:

Users can demonstrate identify attributes (such age) via zero-knowledge proofs (ZKPs) without revealing extraneous information. By preventing sensitive user credentials from being publicly available, encrypted off-chain storage lowers security threats. Smart Contract-Based Verification: To verify the legitimacy of credentials without disclosing personal information, verifiers communicate with smart contracts set up on a blockchain network.

4.4 Secure Credential Revocation and Issuance:

Credential Issuance: Only verified issuers will sign and issue credentials in order to guarantee their integrity and authenticity.

Revocation System: A blockchain-based registry will allow for real-time credential validity checks, allowing for revocation in the event of misuse or expiration.

4.5 Development and Testing:

Development of Mobile Wallets: The wallet will be developed as a cross-platform, decentralized application (Dapp)using React Native.

Blockchain Integration: The solution will be tested using platforms such as Hyperledger Indy and Ethereum-based identification frameworks.

Evaluations of security and performance:

Privacy Testing: Assessing ZKP-based selective disclosure techniques. Scalability analysis entails evaluating performance under varied loads. User experience testing evaluates the usability and efficacy of identity verification.

Chapter 5

OBJECTIVES

The development of a privacy-focused mobile wallet for Self-Sovereign Identity (SSI) management that guarantees user autonomy, security, and interoperability is the main goal of this project. The following are the main goals of the suggested solution:

5.1 Create and implement a Decentralized Identity Wallet:

a mobile wallet that lets users create, save, and control their digital identities safely.

To improve interoperability, make sure that W3C Verifiable Credentials (VCs) and Decentralized Identifiers (DIDs) standards are followed.

5.2 Permit Identity Sharing While Preserving Privacy :

Use Zero-Knowledge Proofs (ZKPs) to enable users to verify identify attributes (such age) without disclosing needless personal information.

Present Selective Disclosure, which lets users reveal only particular aspects of their identities rather than their complete login credentials.

5.3 Create a Safe and Untrustworthy Verification System:

Use blockchain technology to develop a decentralized identity verification system that is not dependent on centralized entities.

Utilize smart contracts to authenticate verifiable credentials while maintaining anonymity.

5.4 Implement Secure Credential Issuance and Revocation:

Permit reliable authority to safely issue digitally signed credentials, such as banks, government institutions, and organizations.

Establish a real-time credential revocation system to deactivate compromised or outdated credentials.

5.5 Ensure Data Security and User Control:

Keep identity credentials in secure, off-chain encrypted storage to prevent exposure on public blockchain networks. Give users total ownership and control over their personal data to stop unauthorized access by third parties.

5.6 Boost Performance and Scalability:

Examine and integrate suitable blockchain technologies (such Hyperledger Indy, Ethereum, or Polygon) to ensure efficient and scalable operations.

To find out how responsive the system is to different workloads, test its performance.

5.7 Confirm Compliance with Privacy Regulations and Interoperability:

Align the system with global privacy laws such as the CCPA and GDPR to guarantee legal compliance.

Ensure compatibility with existing digital identity systems to facilitate seamless platform integration.

Chapter 6

SYSTEM DESIGN & IMPLEMENTATION

Building a privacy-preserving mobile wallet for Self-Sovereign Identity (SSI) management requires a methodical approach that combines blockchain technology, cryptographic security mechanisms, Verifiable Credentials (VCs), and Decentralized Identifiers (DIDs). In order to ensure security, user control, and privacy, this section describes the system design, key components, and implementation strategy.

6.1 System Architecture:

The mobile wallet's decentralized identity method ensures user autonomy, security, and privacy. The architecture consists of three main parts:

6.1.1 Problems with Identity :

Trusted entities (government agencies, financial institutions, and academic institutions) issue digitally signed Verifiable Credentials (VCs).

Use Decentralized Identifiers (DIDs) to create credentials that are impermeable and verifiable.

6.1.2 Identity Owners (Users):

Individuals store their DIDs and Verifiable Credentials in their mobile wallets.

Keep complete control over their identification data and use Selective Disclosure strategies to share exactly the information that is required.

6.1.3 Identity Verifiers:

Organizations or individuals that authenticate users by confirming their credentials.

Avoid the requirement for centralized storage by employing blockchain-based methods to verify authenticity.

6.2 Basic Components of the System:

The implementation of the system consists of several crucial components:

6.2.1 Administration of DIDs:

The DIDs created by users are protected by cryptographic key pairs.
DIDs are stored on a distributed ledger, which ensures transparency and resistance to manipulation.

6.2.2 Secure storage is provided for VCs:

Digitally signed credentials from issuers are securely stored in users' mobile wallets.
Credentials are encrypted and stored off-chain to avoid unwanted access.

6.2.3 Blockchain Identity Verification:

Smart contracts built on the blockchain validate credentials without revealing personal data.
A revocation registry that updates credential status in real time prevents the use of expired or revoked credentials.

6.2.4 Zero-Preserving Authentication:

verify specific parts of their identities, such as age, using Knowledge Proofs (ZKPs) without disclosing unnecessary information.
By guaranteeing that users only divulge information that is required, Selective Disclosure enhances privacy.

6.3 The Method of Implementation:

The mobile wallet is developed through a systematic design, development, and testing process to ensure effectiveness and security.

6.3.1 The technological stack:

A front-end development tool called React Native is used to make mobile apps that work on several platforms. Backend development is the use of Node.js and Express.js to enable secure communication.

Blockchain Integration: Ethereum or Hyperledger Indy-based decentralized identity verification. To ensure data protection, security techniques like AES encryption and Zero-Knowledge Proofs are used.

6.3.2 Workflow of the User Registration System:

Users create a Decentralized Identifier, which is securely stored in their mobile wallet.

Credentials Issue

A digitally signed Verifiable Credential from a verified issuer is given after identification has been verified. Credential Storage Credentials are securely stored off-chain even if a cryptographic hash is stored on the blockchain .The Verification Process

Users check their credentials using Selective Disclosure or Zero-Knowledge Proofs.

Smart contracts verify that the credential is legitimate

6.3.3 Improvements to Security and Privacy:

Decentralized identity management eliminates the need for centralized data storage.

Encrypted off-chain storage protects sensitive information.

Real-time credential validity verification is made possible by a blockchain-based revocation system.

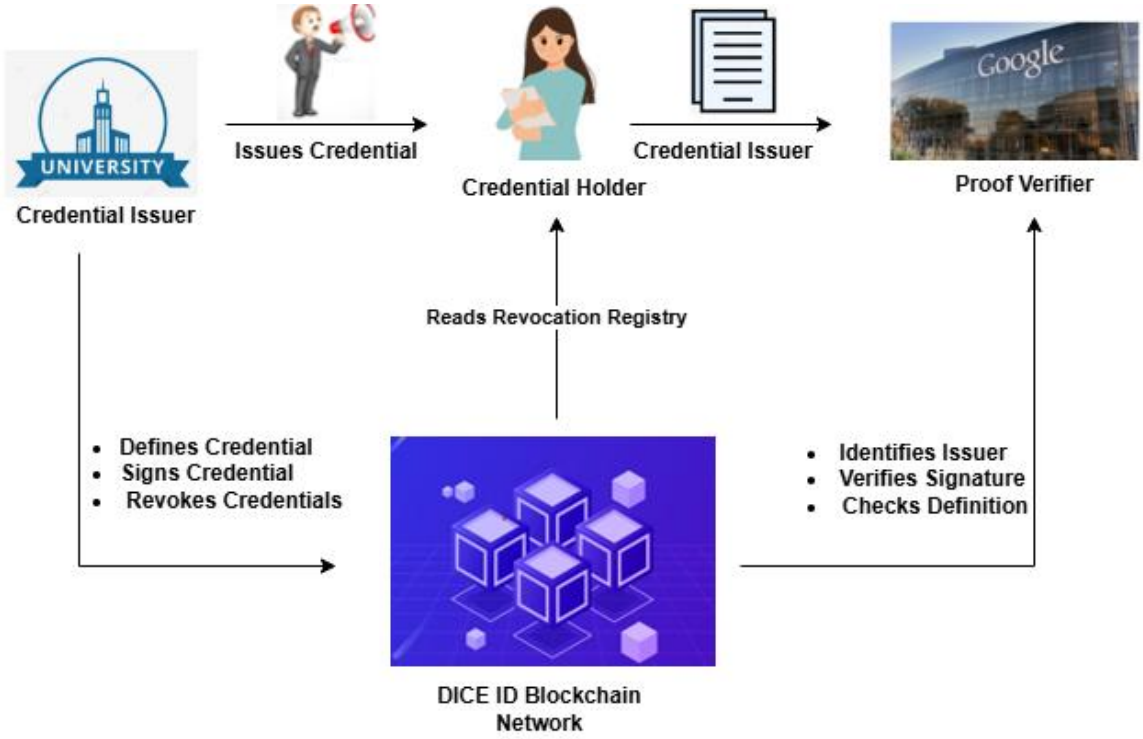


Figure 6.1: ***DICE ID Credential Flow – Blockchain-based Identity System***

Overview of the DICE ID Credential Flow

The DICE ID Blockchain Network is used to issue, store, and verify digital credentials, as the diagram illustrates.

An issuer, such as a university, creates credentials, signs them, and adds **metadata to the** blockchain.

Credentials are received by the holder (user) and saved in a digital wallet.

Verifier (e.g., Employer): Asks for proof and confirms it by looking up the issuer's identity, signature, and blockchain revocation status.

Blockchain Network: Provides revocation tracking, data integrity, and trust without disclosing personal information. Digital identity sharing that is user-controlled, safe, and verifiable is made possible by this technology.

Component	Technology Used	Function
Identity Representation	Decentralized Identifiers (DIDs)	Enables user identification in a decentralized manner without central control.
Credential Format	Verifiable Credentials (VCs)	Facilitates secure, verifiable sharing of identity-related data.
Blockchain Platform	Hyperledger Indy / Ethereum	Used for storing public identity references and credential metadata.
Privacy Mechanism	Zero-Knowledge Proofs (ZKPs)	Allows users to prove information without exposing actual data.
Mobile Wallet Framework	React Native / Flutter	Supports the development of a portable identity wallet for mobile platforms.

Table 6.1: Utilized Technology

Decentralized Identifiers (DIDs):

DIDs enable individuals to establish and control their own digital identities independently, without depending on centralized authorities, ensuring personal ownership of identity data.

Verifiable Credentials (VCs):

VCs are digital attestations that securely convey information about a person, such as qualifications or skills, allowing others to verify these claims reliably and securely.

Blockchain Platform (Hyperledger Indy / Ethereum):

These blockchain networks provide a secure and immutable system to store identity references and credential data, making sure records are transparent and resistant to tampering.

Zero-Knowledge Proofs (ZKPs):

ZKPs allow a person to prove the truth of a statement (like verifying age or membership) without revealing any additional personal details, enhancing privacy during identity checks.

Mobile Wallet Framework (React Native / Flutter):

These frameworks support building mobile applications that let users securely store and manage their digital identity credentials on smartphones, facilitating easy access and sharing.

Chapter 7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

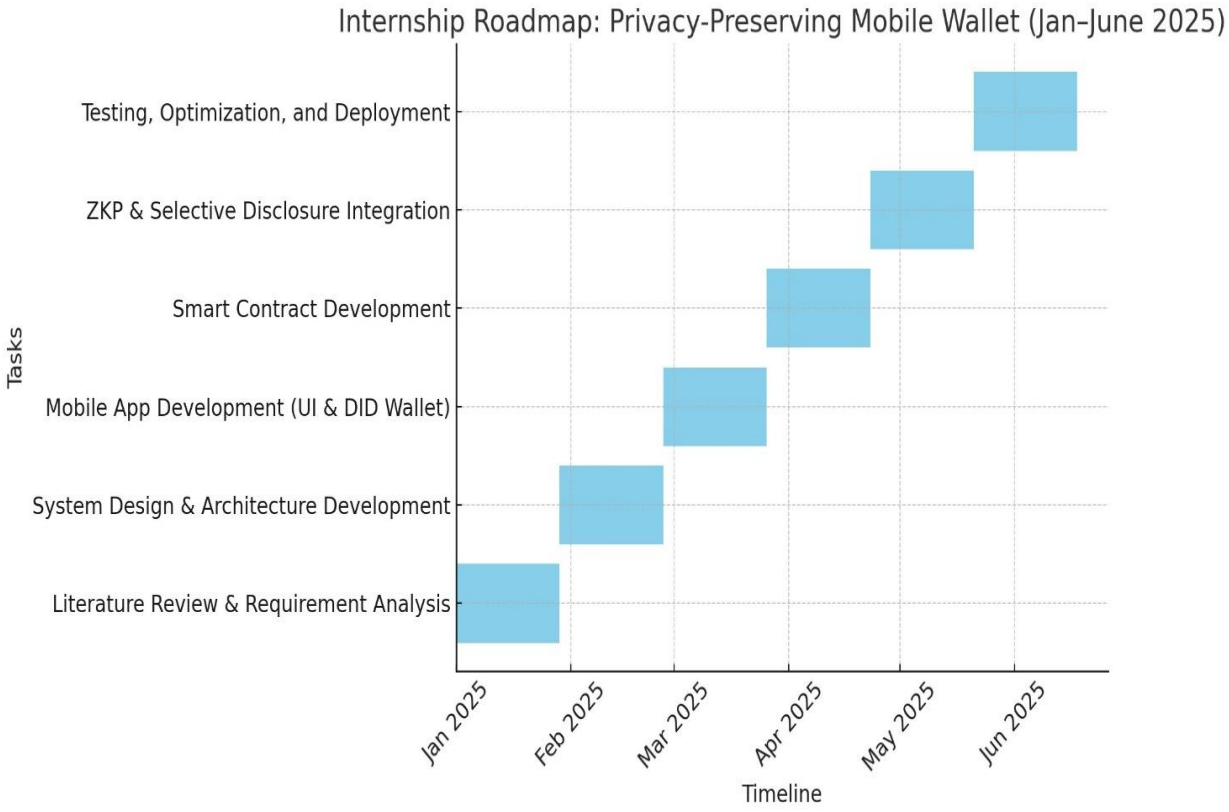


Figure 7.2: Gantt chart

Internship Roadmap (Jan–June 2025)

Jan: Study existing solutions and define needs

Feb: Plan system layout and architecture

Mar: Build app interface and connect DID wallet

Apr: Develop and deploy smart contracts

May: Add privacy features using ZKPs

Jun: Test the system and launch

Chapter 8

OUTCOMES

- The Wipro DICE ID platform was used to create a decentralized mobile wallet that allows for the safe and private administration of digital IDs.

- Users had complete control over their personal information while receiving, storing, and sharing Verifiable Credentials from accredited organizations.

- To improve privacy, the system implemented selective disclosure, which permits users to divulge only the essential portions of their login credentials.

- A tamper-proof method for confirming credentials without disclosing private information was made possible by blockchain integration.

- Easy onboarding, an OTP-based login, and user-friendly interface navigation all contributed to a flawless user experience.

- The wallet ensured wide compatibility and future scalability by adhering the W3C standards for Verifiable Credentials and Decentralized Identifiers.

- The research demonstrated the practical applicability of such privacy-focused identity systems in fields such as public services, healthcare, and education.

Chapter 9

RESULTS AND DISCUSSIONS

The efficacy of the Wipro DICE ID-based Self-Sovereign Identity (SSI) mobile wallet in terms of safe identity management was examined and tested. The system effectively demonstrated the user-centric, private, and verifiable nature of decentralized identity handling.

Important Results:

Issuing and storing credentials:

Through the DICE ID platform, users could obtain digital credentials from approved organizations, such as demo issuers.

Only the user had secure access to these credentials, which were kept in the wallet.

Sharing While Preserving Privacy:

The wallet ensured that users' privacy was maintained during identity verification by allowing them to reveal specified portions of their credentials.

The risk of oversharing private information was decreased by this selective disclosure.

Confirmation & Confidence:

Without gaining access to the user's personal information, verifiers could verify the legitimacy of credentials.

Credential validity, including revocation status and signature verification, was verified via the blockchain network.

Easy Onboarding of Users:

From OTP-based login to credential acceptance and identity verification, the system provided a smooth experience.

User understanding and confidence were increased by email confirmations and simple interface navigation.

Safety and Openness:

Credential verification gained a transparent and impenetrable layer with the incorporation of blockchain technology.

Throughout the process, users had complete control over their identifying data.

Discussion:

The research demonstrates that using mobile platforms to deploy a decentralized, privacy-preserving identity system is feasible. It demonstrates that SSI can be implemented practically with DICE ID and other current technologies. Because it is decentralized, users are not reliant on any one entity, which enhances data security and ownership.

This method has a lot of promise for use in fields where privacy and trust are essential, like healthcare, education, and government. Such systems may soon become commonplace in digital interactions due to their user-friendly design and adherence to contemporary digital identity requirements.

Chapter 10

CONCLUSION

This project offers a secure, privacy-focused mobile wallet for managing self-sovereign identities (SSI). Thanks to blockchain technology and verifiable credentials, the wallet allows users to store and share credentials without relying on centralised services, giving them total control over their digital identities.

Users can receive and manage credentials within a secure app thanks to the integration of Wipro DICE ID, and features like consent-based sharing and selective disclosure guarantee that only pertinent information is exposed. The process is made even more secure and trustworthy by the use of digital signatures and revocation procedures.

Overall, this strategy offers scalable and interoperable identity solutions that can be applied in a variety of industries, including government, healthcare, and education, in addition to improving user privacy and control. It represents a gradual move toward digital identification systems that are user-centric and decentralized.

REFERENCES

[1] Armando, A., Carbone, R., Compagna, L., Cuellar, J., & Tobarra, L. (2008). An analytical study of SAML 2.0-based single sign-on vulnerabilities: A case study involving Google Apps. In Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering (pp. 1–10). [1]

[2] Shehu, A.-S., Pinto, A., & Correia, M. E. (2020). Secure delegation of access rights in identity management systems. In Proceedings of the 17th International Joint Conference on e-Business and Telecommunications – SECRYPT (pp. 638–644). SciTePress. [2]

[3] Ribeiro, C., Leitold, H., Esposito, S., & Mitzam, D. (2017). Stork: A real-world, large-scale, and diverse eID management framework. International Journal of Information Security, 1–17. [3]

[4] Makhdoom, I., Zhou, I., Abolhasan, M., Lipman, J., & Ni, W. (2020). PrivySharing: A blockchain-enabled architecture for secure and privacy-aware data exchange in smart cities. Computers & Security, 88, 101653. [4]

[5] Rantos, K., Drosatos, G., Kritsas, A., Ilioudis, C., Papanikolaou, A., & Filippidis, A. P. (2019). A consent management solution for personal data in IoT using blockchain technology. Security and Communication Networks, 2019. [5]

[6] Lyons, K. T. T., & Courcelas, L. (2019). Blockchain and Digital Identity. The European Union Blockchain Observatory and Forum, Thematic Report v1.0, May. [6]

[7] Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., & Mortimore, C. (2014). OpenID Connect Core 1.0 including errata set 1. The OpenID Foundation. Retrieved November 17, 2022, from https://openid.net/specs/openid-connect-core-1_0.html [7]

[8] Srinivas, S., Balfanz, D., Tiffany, E., Czeskis, A., & FIDO Alliance. (2015). An overview of Universal 2nd Factor (U2F) security protocols. FIDO Alliance Proposed Standard, 15. [8]

[9] Liu, Y., He, D., Obaidat, M. S., Kumar, N., Khan, M. K., & Choo, K.-K. R. (2020). A comprehensive review of identity management using blockchain technology. Journal of Network and Computer Applications, 166, 102731. [9]

[10] Vasilescu, A. (2017). The current state and future direction of eID under eIDAS: Timeline and progress overview. October, pp. 1–7. [10]

APPENDIX-A
PSUEDOCODE

Smart Contract Design

- The contract defines a Credential structure that follows the W3C Verifiable Credential model:

```
struct Credential {
    string skill; // Name of the certified skill
    string holderDID; // DID of the credential holder
    string issuerDID; // DID of the credential issuer
    uint256 issuedAt; // Timestamp of issuance
    bool revoked; // Revocation status
    bytes signature; // Issuer's cryptographic proof
}
```

- Each credential is uniquely identified by a credential ID, computed using a keccak256 hash of its attributes. This ensures uniqueness while enabling efficient verification.

skill: Represents the qualification or competency being certified.

Holder DID: The decentralized identifier of the person or entity receiving the credential.

Issuer DID: The decentralized identifier of the authority or organization that issued the credential.

Issued At: Records the exact time when the credential was created.

revoked: Boolean flag to denote if the credential is still valid or has been revoked.

signature: A digital signature from the issuer to validate the authenticity and integrity of the credential data.

Issuer Authorization

A mapping maintains a list of trusted issuers, ensuring that only verified entities can issue credentials:

```
mapping(address => bool) public issuers;
modifier onlyIssuer() {
   require(issuers[msg.sender], "Not an authorized issuer");
   _;
}
```

Issuers can be added or removed dynamically, providing flexibility for governance and compliance with institutional requirements.

- A mapping(address => bool) is used to maintain a list of authorized issuers.
- Each address is marked true if it is allowed to issue credentials.
- The only Issuer modifier checks if the caller (msg. sender) is a verified issuer.
- If the caller is not authorized, the function execution is blocked with an error message.
- This ensures that only trusted entities can issue credentials.
- The contract allows issuer addresses to be added or removed dynamically.
- This flexible design supports ongoing governance, policy changes, and secure access control.

Credential Issuance

The issuance process involves the issuer generating a cryptographic signature and storing only the metadata on-chain:

```
function issueCredential(
   string memory _skill,
   string memory _holderDID,
   string memory _issuerDID,
   bytes memory _signature
) public onlyIssuer returns (bytes32)
```

```
{
bytes32 credentialId = keccak256(abi.encodePacked(_skill, _holderDID, _issuerDID, block.timestamp));
    credentials[credentialId] = Credential({
        skill: _skill,
        holderDID: _holderDID,
        issuerDID: _issuerDID,
       issuedAt: block.timestamp,
        revoked: false,
        signature: _signature
    }
);
    emit CredentialIssued(credentialId, _skill, _holderDID, _issuerDID, block.timestamp);
    return credentialId;
}
```

This approach ensures that credentials remain tamper-proof while minimizing on-chain storage costs.

The function allows verified issuers to issue credentials securely using the onlyIssuer modifier. It takes the skill name, holder's DID, issuer's DID, and a digital signature as inputs.

A unique credential ID is created using keccak256, which hashes these inputs along with the current timestamp.

This unique ID helps ensure that each credential is distinct and cannot be duplicated or tampered with.

The credential metadata (skill, DIDs, timestamp, revocation status, and signature) is stored on-chain. Full credential data is not stored, reducing blockchain storage costs.

An event is emitted after successful issuance for tracking and external monitoring. This method ensures both the authenticity of credentials and efficient use of blockchain resources.

Credential Verification

Verification is performed using ECDSA signature validation, allowing any verifier to confirm the credential's authenticity:

```
function verifyCredential(bytes32 _credentialId, address _issuer)
    public view returns (bool isValid, bool isRevoked)

 {
    require(credentials[_credentialId].issuedAt != 0, "Credential does not exist");
    Credential memory cred = credentials[_credentialId];

 bytes32 hash = keccak256(abi.encodePacked(cred.skill, cred.holderDID, cred.issuerDID,
cred.issuedAt));

 bool validSignature = recoverSigner(hash, cred.signature) == _issuer;
    return (validSignature, cred.revoked);
}
function recoverSigner(bytes32 hash, bytes memory signature) internal pure returns (address)
{
    bytes32 r;
    bytes32 s;
    uint8 v;




assembly {
        r := mload(add(signature, 32))
        s := mload(add(signature, 64))
       v := byte(0, mload(add(signature, 96)))
```

```
 }
   return ecrecover(hash, v, r, s);
}
```

The verification mechanism ensures that credentials are validated without relying on the issuer, making the system decentralized and trust l ss.

The verification function ensures the authenticity of a credential by using ECDSA signature validation. It first checks that the credential exists by confirming the issuance timestamp is present. Then, it recreates a hash of the credential's important data—such as the skill, holder DID, issuer DID, and issuance time—using the same hashing method applied during the credential issuance. The recover Signer function extracts the signer's address from the digital signature by separating it into components and applying the recover operation.

This recovered address is then compared with the expected issuer's address to verify the signature's validity. The function also returns the revocation status of the credential. By enabling anyone to independently validate the credential on-chain without depending on the issuer, this approach supports a decentralized and trust less verification system, enhancing security and user trust.

Credential Revocation

Credentials can be revoked by the issuer when necessary, such as in cases of credential expiration or policy updates. The revocation status is stored as a Boolean flag:

```
function revokeCredential(bytes32 _credentialId) public onlyIssuer {
   require(credentials[_credentialId].issuedAt != 0, "Credential does not exist");
   require(keccak256(abi.encodePacked(credentials[_credentialId].issuerDID))

 == keccak256(abi.encodePacked(msg.sender)), "Only the issuer can revoke");
   credentials[_credentialId].revoked = true;
   emit CredentialRevoked(_credentialId, credentials[_credentialId].issuerDID);
}
```

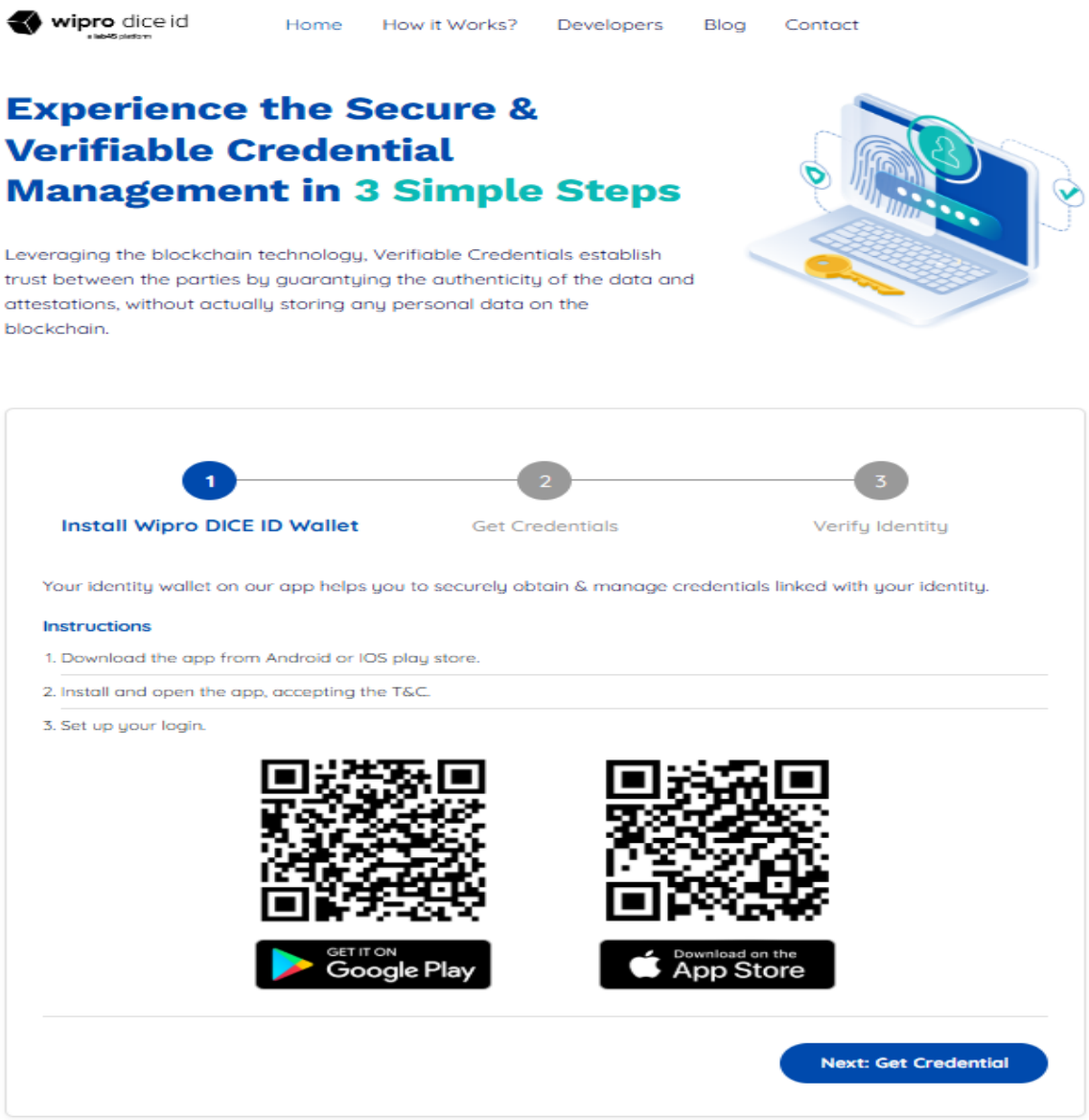
Since blockchain data is immutable, the credential itself cannot be deleted, ensuring a transparent audit history.

Issuers have the ability to revoke credentials when necessary, such as due to expiration or changes in policies. This is achieved by updating a Boolean flag in the credential's record to indicate it is no longer valid. The revocation function first checks that the credential exists and confirms that the caller is the original issuer by matching the issuer's decentralized identifier with the caller's address.

Since data stored on the blockchain cannot be erased or modified, the credential itself remains permanently recorded. Instead of deleting the credential, marking it as revoked preserves an immutable and transparent audit trail. This ensures that the system maintains accountability and allows anyone to verify the current status of a credential without losing historical information.

APPENDIX-B

SCREENSHOTS



Screenshot 1: Install the Wipro DICE ID Wallet first.

The DICE ID app is the first thing you download to your phone.

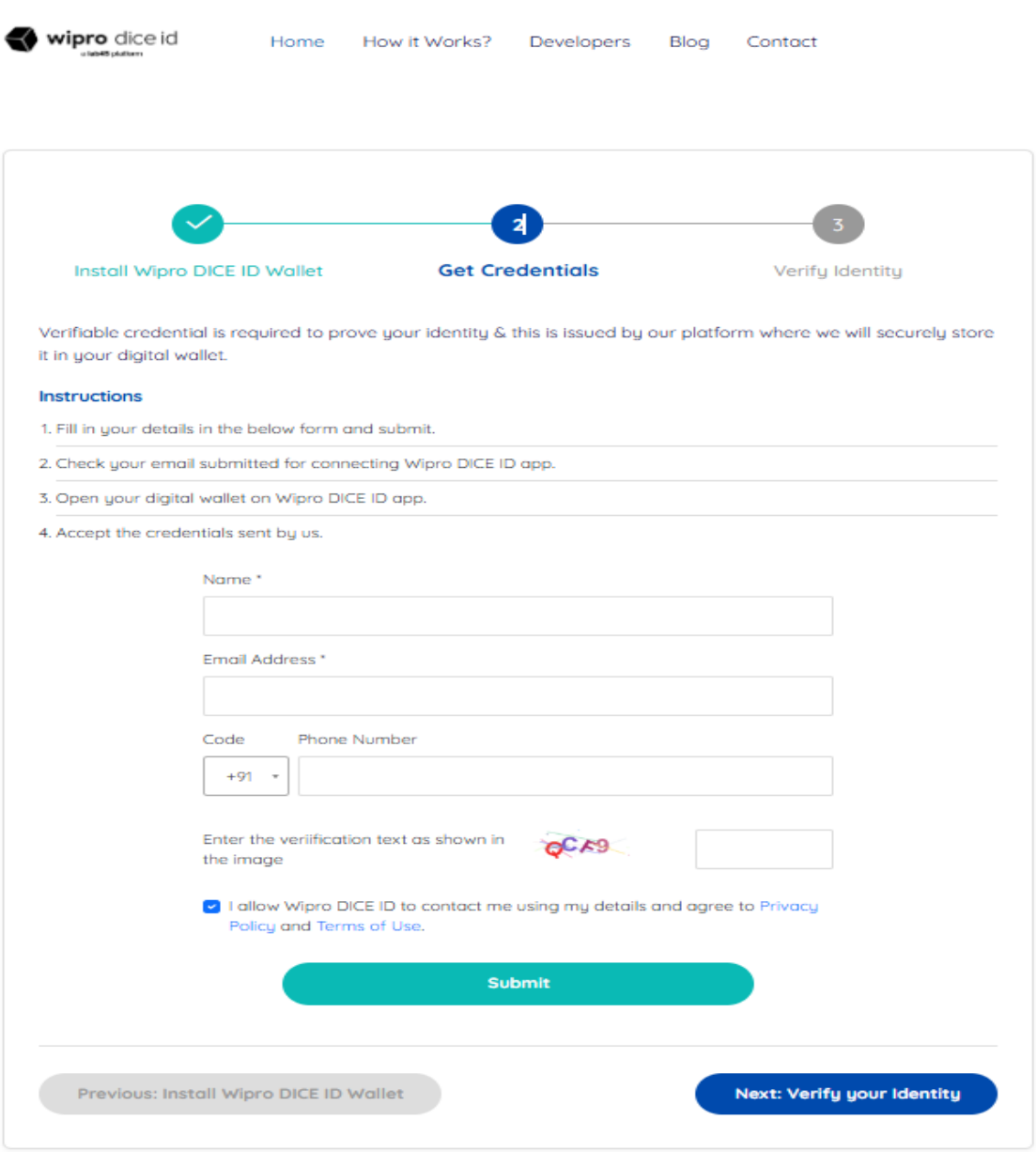
To store and manage credentials, the app functions as your identity wallet.

- Directions: Download the app (iOS and Android downloads are provided below).

  Accept the terms and conditions.

  Configure your login, which is typically a passcode or security key.

📷 QR codes: Scan the Google Play Store QR code on the left.

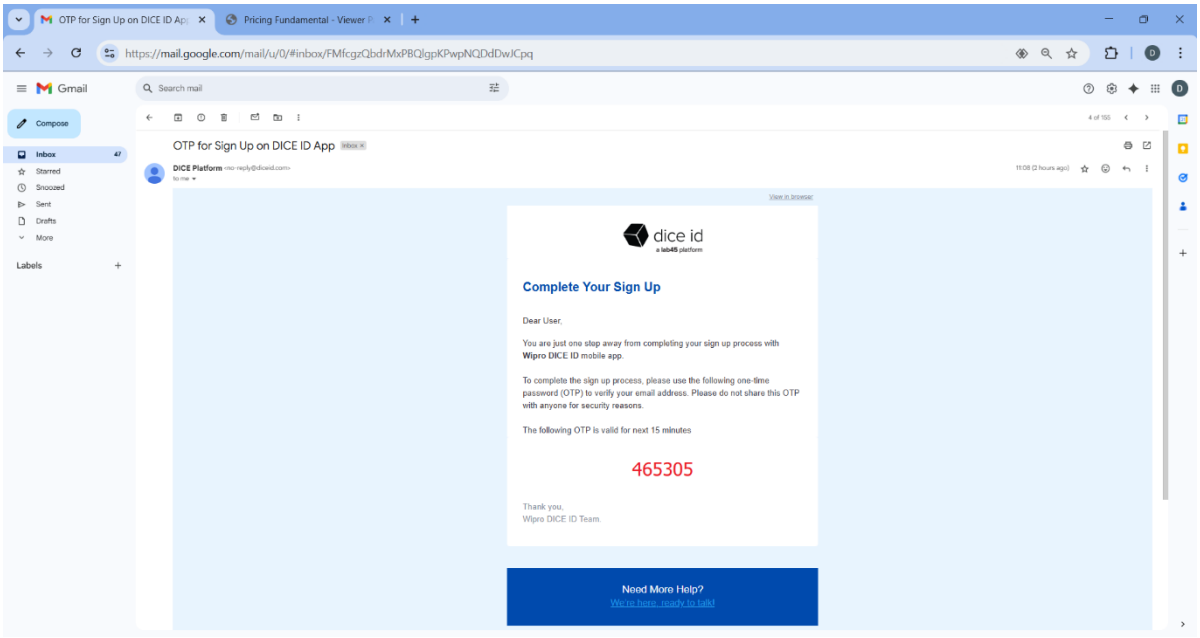


Screenshot 2: Obtain Wipro DICE ID Credentials

- The user is guided by this screen to safely obtain their Verifiable Credential.
- 📝 Directions: Enter your name, email, phone number, and CAPTCHA.
- Look for a link from Wipro DICE ID in your email.
- Accept the credentials after opening the app.
- Your wallet is where your credentials are kept.
- 📌 Goal: To provide a blockchain-based identity verification credential.

Buttons:

- Send information by submitting
- Next, proceed to Step 3: Identity Verification.



Screenshot 3: Wipro DICE ID Sign-Up OTP Email

The user received this email confirmation during the sign-up procedure from the Wipro DICE ID platform.

📩 Email Details: The DICE ID Team sent it.

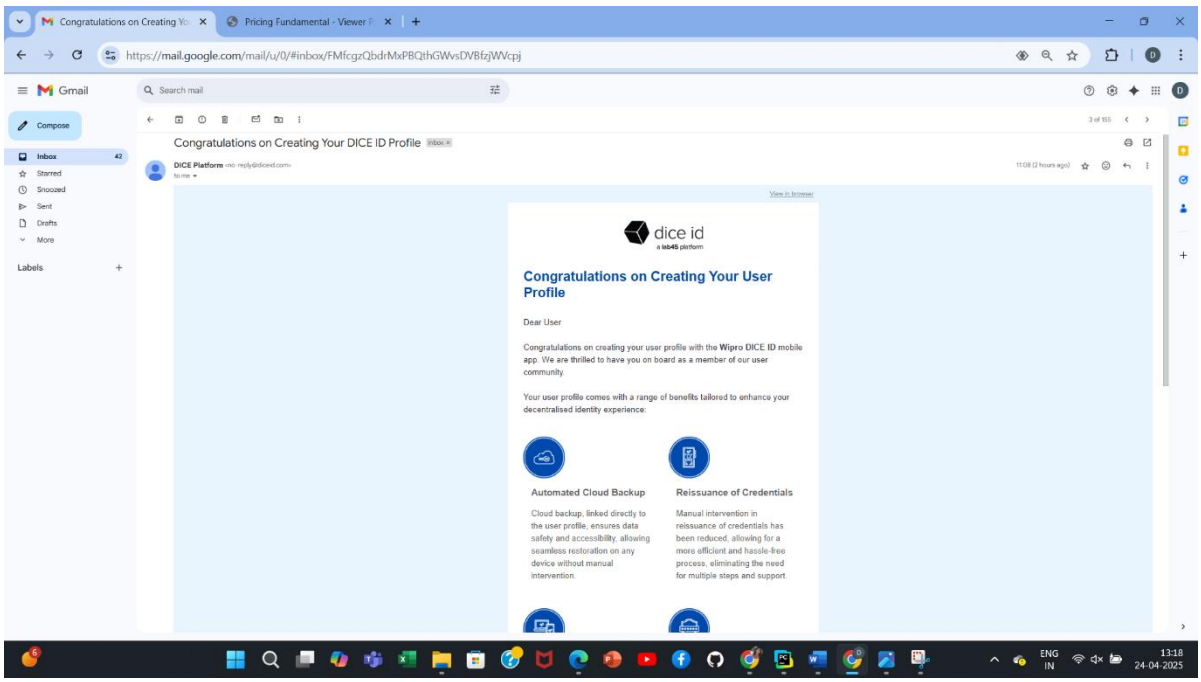
Subject: OTP for DICE ID App Sign-Up

Content: notifies the user that they are just one step away from finishing the registration process.

gives a six-digit One-Time Password (OTP): 465305.

The OTP is good for fifteen minutes.

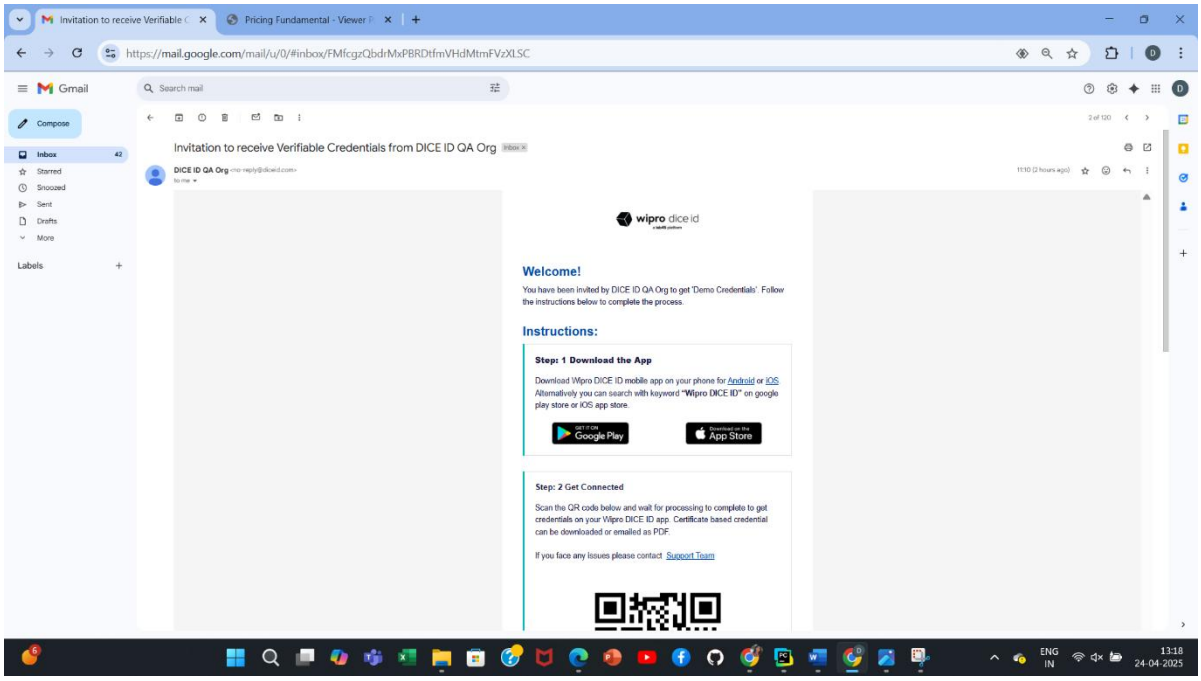
🔐 Purpose: To verify the user's email during registration for secure identification setup on the Wipro DICE ID app.



Screenshot 4:DICE ID Confirmation of User Profile Creation

After a user completes their profile creation on the mobile app, Wipro DICE ID sends a confirmation email to verify that the registration process was successfully finished. This message confirms that the user's profile has been properly established and usually contains important information like the user's unique ID, instructions on navigating the app, and advice on maintaining account security.

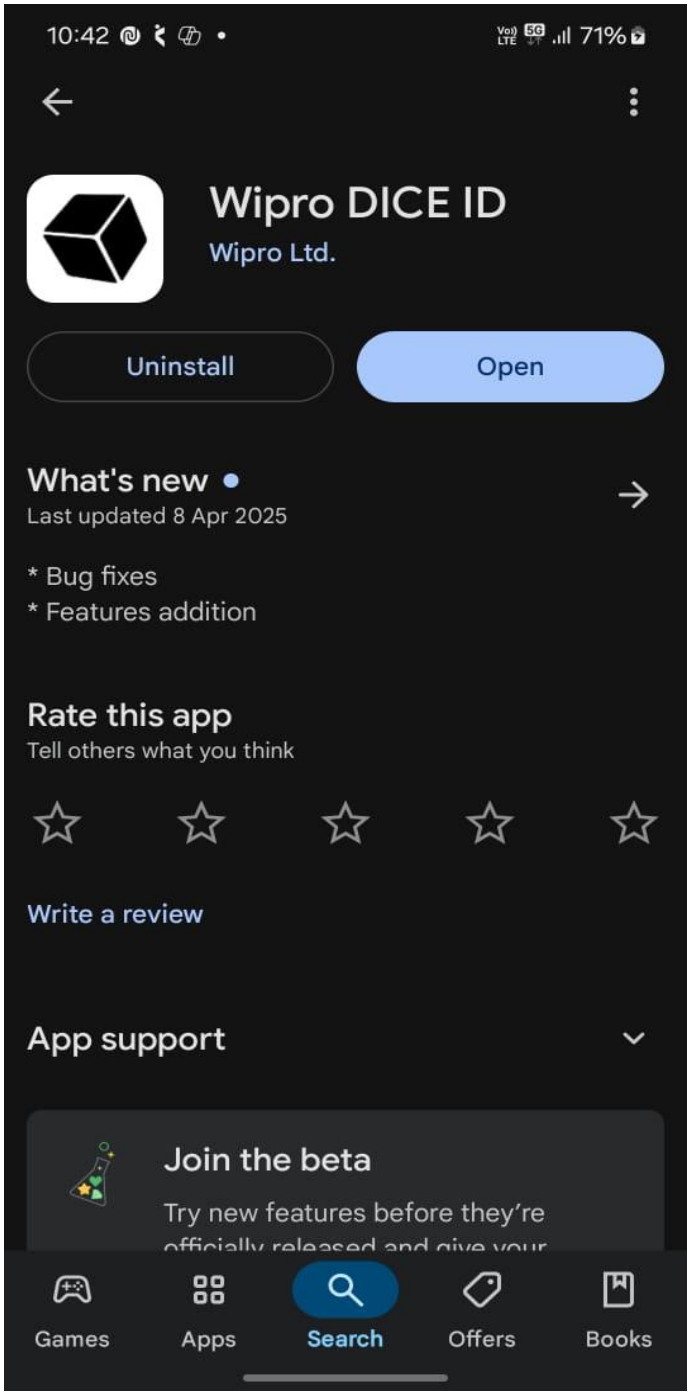
The email may also provide guidance on updating personal information, linking additional credentials, or adjusting privacy preferences. It often includes support contact details for any questions or assistance the user might need. This confirmation helps users feel assured that their identity has been correctly registered and supports them in getting started with the app's features.



Screenshot 5: issue: verifiable digital credentials

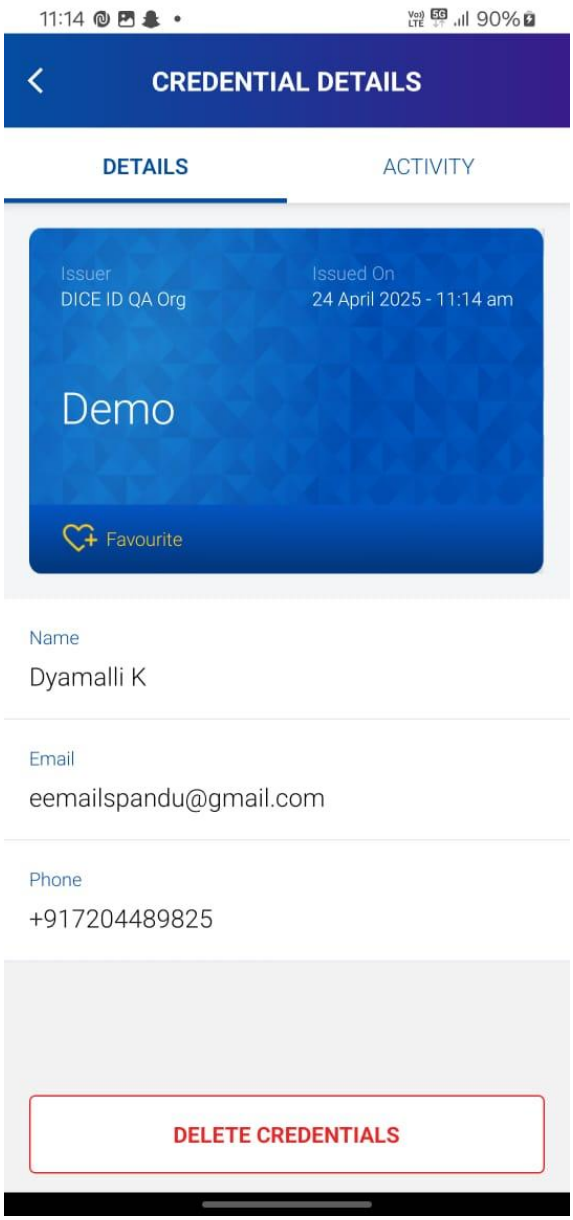
This email serves as an official digital invitation for the user to join the Wipro DICE ID platform and begin the process of obtaining verifiable digital credentials. The onboarding uses secure QR codes alongside mobile technology to create a direct and protected link between the user's digital identity and the credential issuer.

By scanning the QR code with their mobile device, users establish a trusted connection that ensures the credentials issued are genuine and specifically tied to their identity, reducing risks of fraud. This method makes the process easy and secure, allowing users to receive, manage, and present their digital credentials conveniently. Overall, it enhances user control over personal data while maintaining strong security throughout the credential issuance process.



Screenshot 6:Install the DICE ID Wallet

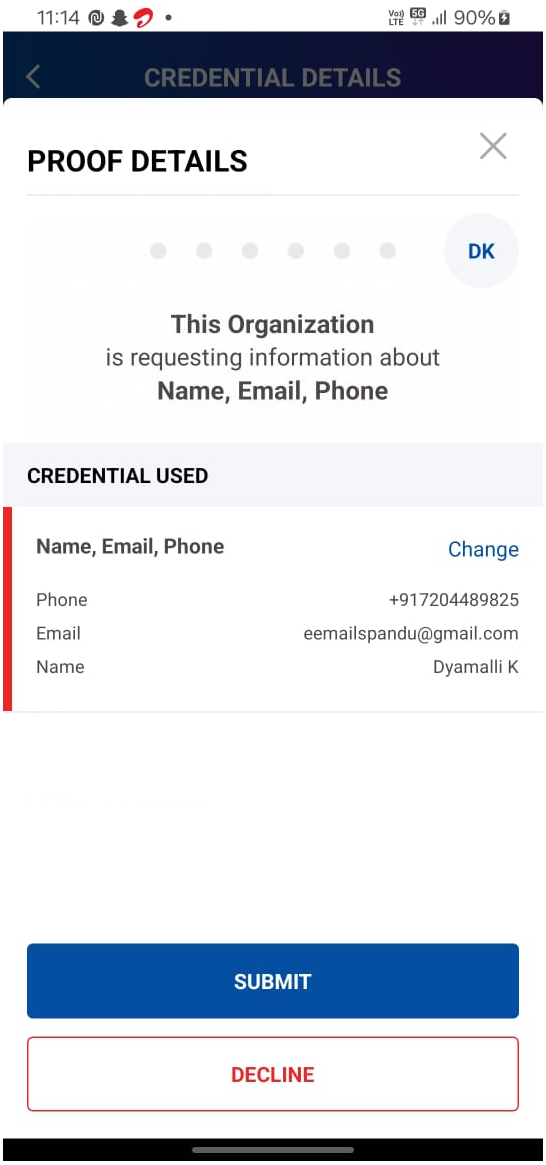
- Download the Wipro DICE ID app from the Android or iOS store.
- Install and open the app, accept the terms and conditions, and set up your login credentials.



Screenshot 7:self-sovereign identity

2.This screenshot is part of a self-sovereign identity (SSI) system, where:

- Users are the sole owners of their identity data.
- Credentials are stored locally in the mobile wallet, not on a centralized server.
- Users can choose when and to whom to disclose their information.
- This approach enhances privacy, security, and user control.



Screenshot 8: Sharing Credentials with Wipro DICE ID

The application asks the user to provide an organization with their verified name, email address, and phone number.

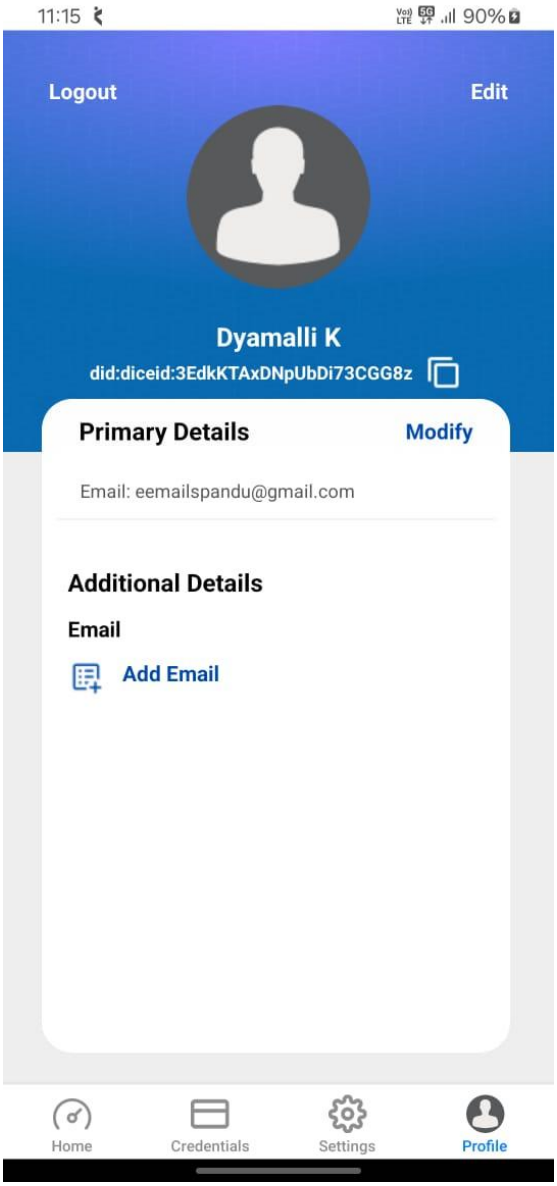
Name: Dyamalli K.

eemailspandu@gmail.com is the email.

Contact number: +91 7204489825

Options include: Change to update data, Decline to cancel, and Submit to share.

guarantees safe, permission-based data exchange.



Screenshot 9: Profile Screen for Wipro DICE ID

The Wipro DICE ID app's Profile area is displayed in this picture.

Name of user: Dyamalli K

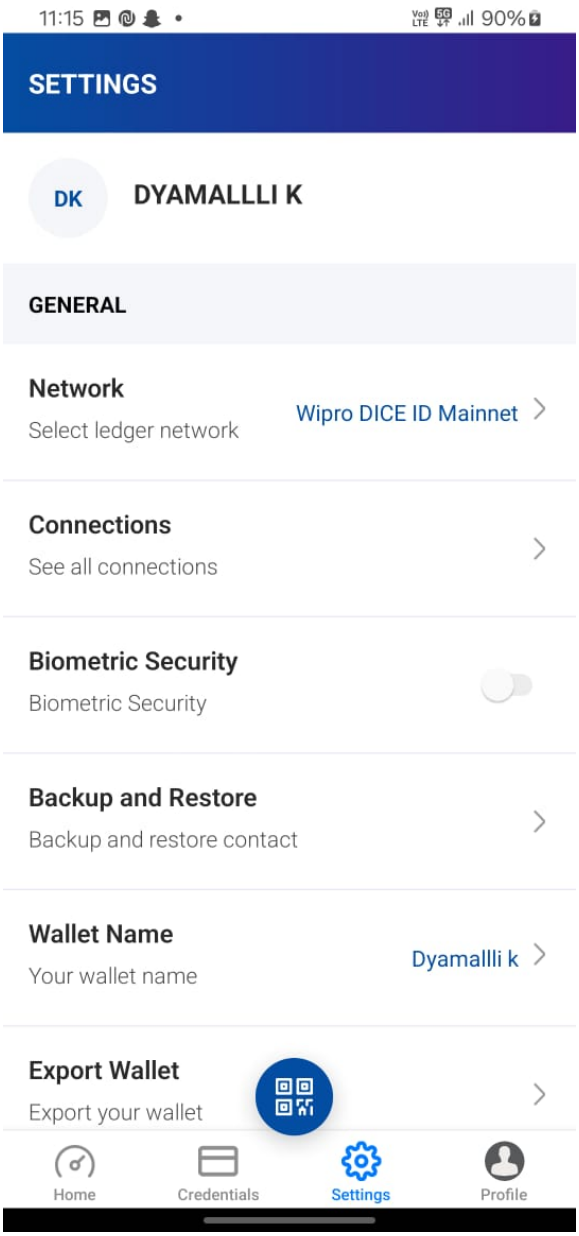
The user's digital identity is represented by their DID (Decentralized Identifier), which is a unique ID (did:diceid:3EdkKTAxDNpUbDi73CGG8z).

The registered email address, eemailspandu@gmail.com, is displayed as the primary detail.

Extra Information: The ability to include other emails.

At the top are buttons to edit your profile or log out.

Included in the navigation bar are Home, Credentials, Settings, and Profile. The user can manage and update their verified identification information within the app on this screen.



Screenshot 10: Screen of Settings for Wipro DICE ID

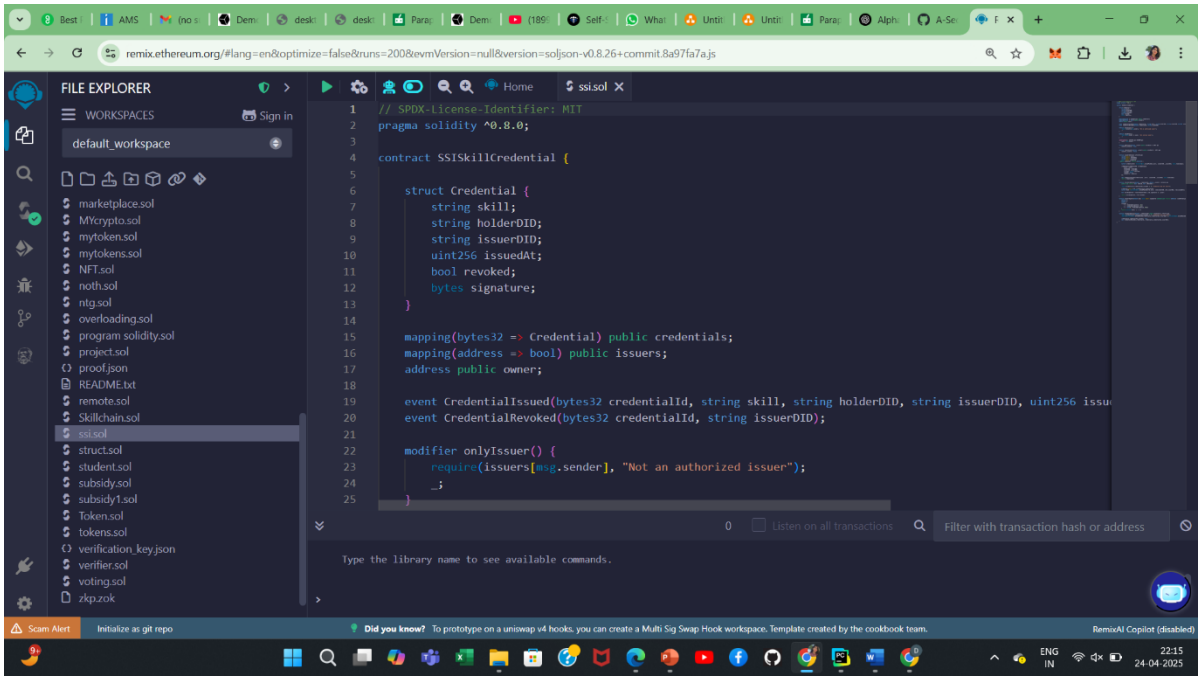
The Wipro DICE ID mobile app's Settings section, where users may control identification wallet settings, is shown in this image.

Important Information: User: Dyamalli K

Network: Linked to the decentralized ledger network known as the Wipro DICE ID Main net.

Connections: The ability to see every link that has been made with organizations.

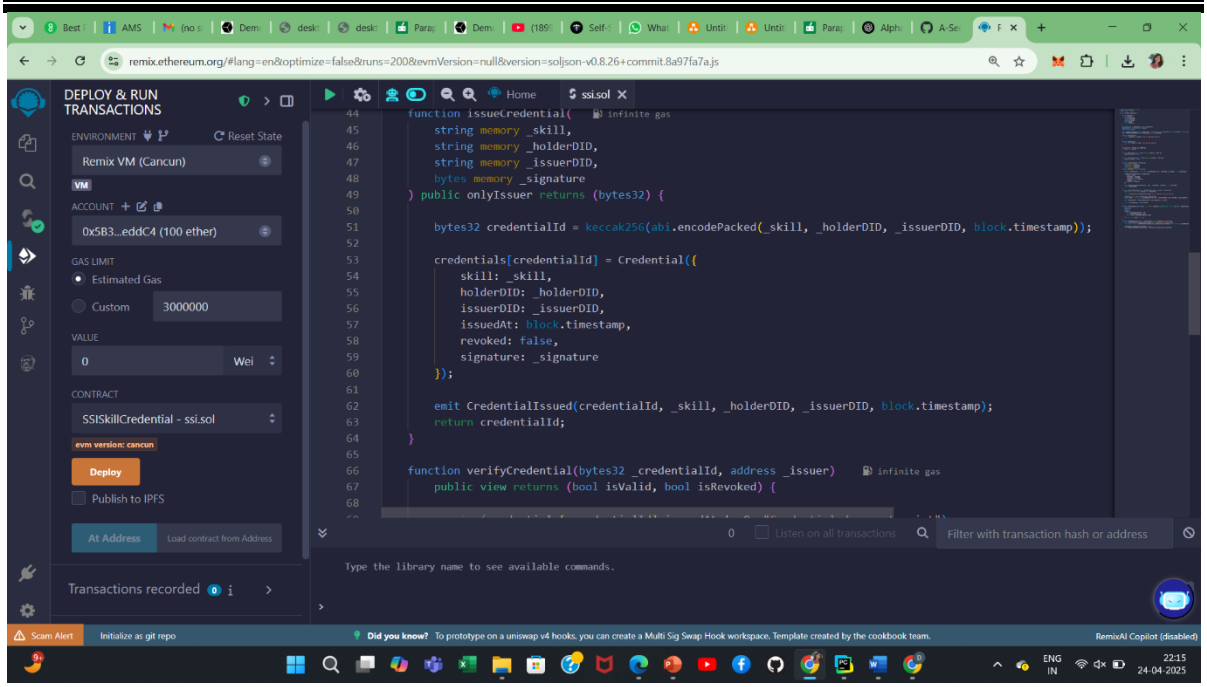
The user may maintain connectivity, security, and backup options for their wallet with verified identification with the use of this page.



Screenshot 11: SSI Skill Credential Contract Implementation using Remix IDE

A unique credential ID is created by applying the keccak256 hashing function to the various attributes of each credential, such as the skill name, holder's DID, issuer's DID, and timestamp. This cryptographic process generates a fixed-length, unique code that changes completely if any of the input data is altered, ensuring the ID is both distinctive and resistant to tampering.

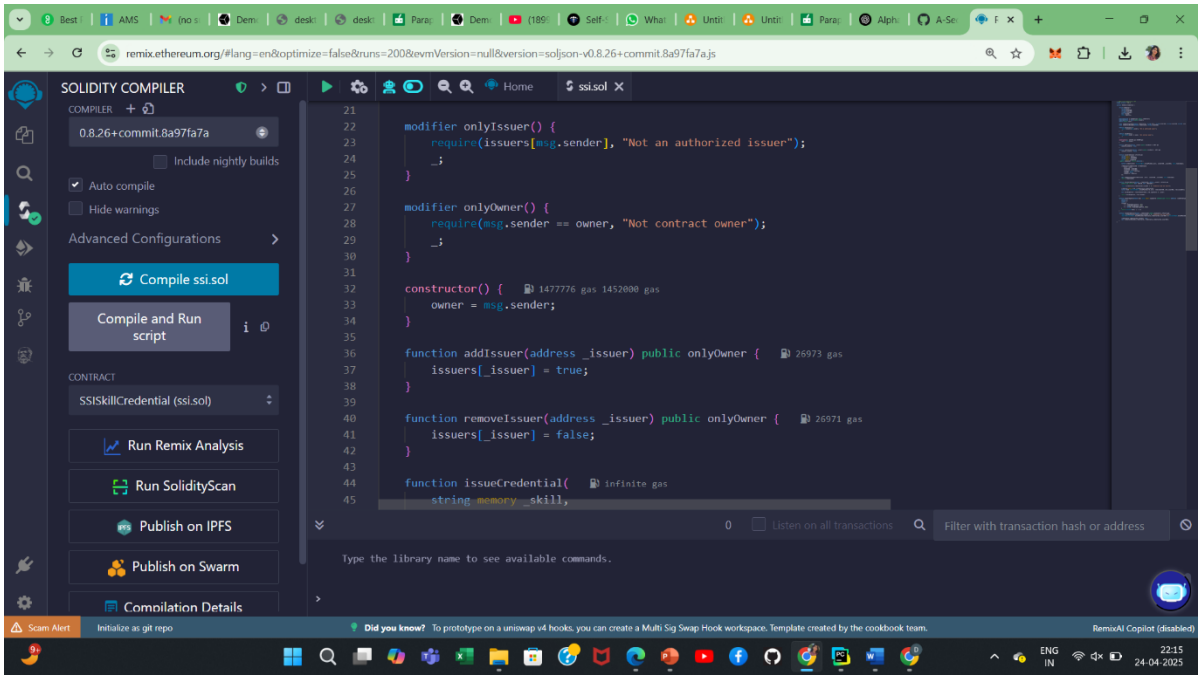
Using keccak256 for credential IDs enables efficient and secure verification since the system only needs to compare these hash values rather than all individual credential details. This technique helps prevent duplication and unauthorized modifications, as any change in the credential's data would produce a different hash, making the credential invalid.



Screenshot 12:Using the Remix IDE to Deploy Smart Contracts and Issue Credentials

The ability to add or remove issuers dynamically offers significant flexibility to the system, enabling it to adapt to evolving governance policies and institutional requirements. This means authorized administrators can update the list of trusted entities allowed to issue digital credentials without redeploying the entire smart contract, which saves time and resources.

This capability is essential for maintaining compliance and security, as it allows the platform to promptly revoke access from issuers who are no longer authorized or have been compromised, while also onboarding new verified issuers smoothly. Managing issuer permissions dynamically helps maintain the credibility and reliability of the credential issuance process. Additionally, it promotes transparency by keeping a clear, auditable record of issuer authorizations over time.



Screenshot 13: SSI Credential Contract Compilation and Access Control Configuration in Remix IDE

Appendix B.13 demonstrates the compilation of a smart contract for Self-Sovereign Identity (SSI) credentials using the Remix IDE. Remix offers an intuitive platform to write, compile, and deploy smart contracts on Ethereum-compatible networks.

The contract enforces role-based access through modifiers like onlyOwner and onlyIssuer, ensuring that certain functions are restricted to specific users.

The onlyOwner modifier restricts administrative tasks—such as managing authorized issuers and transferring ownership—to the contract's deployer, while onlyIssuer limits credential issuance to trusted entities.

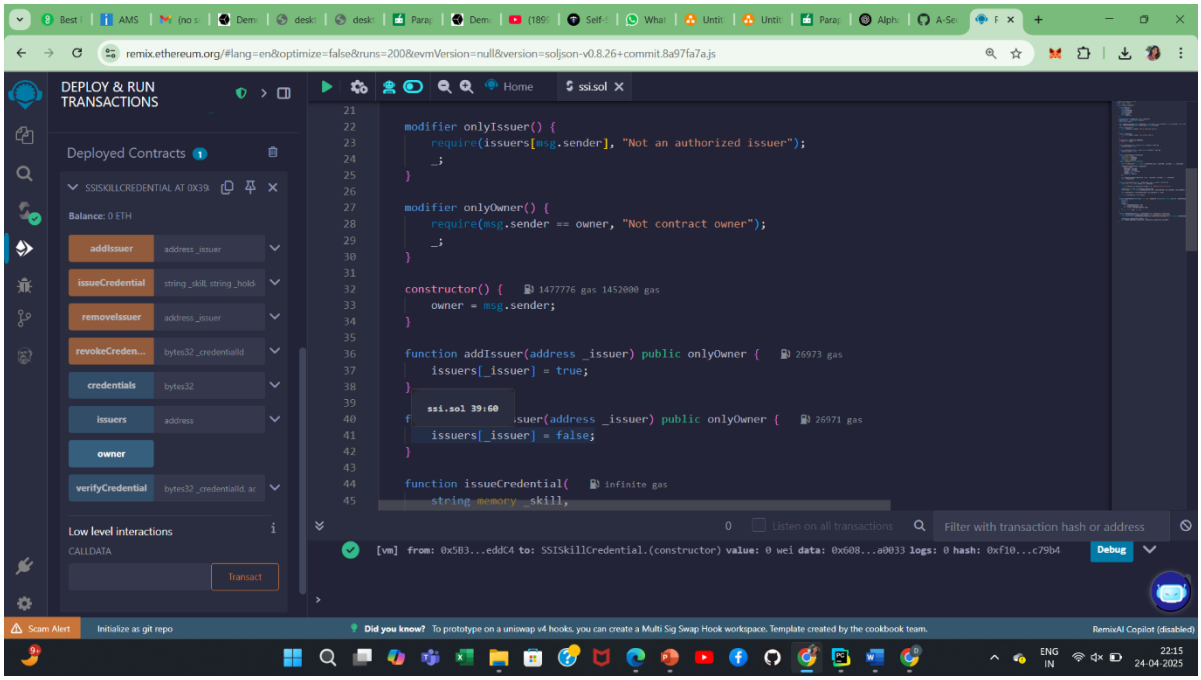
Upon deployment, ownership is automatically assigned to the account that deployed the contract via the constructor, granting that account administrative privileges.

Ownership can later be transferred to another address if necessary, allowing flexible control.

Functions to add or revoke issuers dynamically provide the ability to update who can issue credentials without needing to redeploy the contract, enabling adaptable governance.

On the Remix IDE interface, visible on the left side, compiler settings such as Solidity version 0.8.26 and auto-compile options streamline development by ensuring the code is checked and compiled efficiently during editing.

This configuration ensures secure deployment and effective access management for SSI credential contracts, allowing only authorized parties to issue credentials while providing clear administrative control.



Screenshot 14 :Remix IDE Smart Contract Deployment and Function Testing

This section showcases how the Self-Sovereign Identity (SSI) Skill Credential smart contract is deployed and tested within the Remix IDE environment. The Remix IDE offers a robust platform for developers to compile, deploy, and interact with Ethereum smart contracts in a virtual testing environment.

Once the contract is successfully deployed using the Remix Virtual Machine (VM), various contract functions become available for real-time interaction. These functions—such as addIssuer, issueCredential, and verifyCredential—appear in the left-side panel of the interface, allowing users to test the contract's behavior directly without needing to use an actual blockchain network.

APPENDIX-C
ENCLOSURES

4: High-quality Instruction

Digital credentials allow learners to receive certifications that are secure and easily verified, helping to ensure the credibility of educational achievements.

8: Good Work and Economic Development

These credentials simplify the verification of employment qualifications, improving access to job opportunities and supporting economic progress.

9: Infrastructure, Industry, and Innovation

Blockchain-based digital identity systems provide a modern, reliable foundation for identity management, encouraging technological advancement and innovation.

10: Decreased Inequalities

Portable digital credentials make it possible for individuals from all backgrounds to have equal access to their personal identification and qualifications.

16: Justice, Peace, and Strong Institutions

Digital identity solutions promote better governance by enhancing transparency and building trust between citizens and institutions.

17: Partnerships to Achieve the Goals

Effective digital identity systems require collaboration among organizations, technology providers, and users to successfully meet shared objectives.

Arshiya Lubna - Internship report diya

ORIGINALITY REPORT

18%	9%	10%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	11%
2	Amal Abid, Saoussen Cheikhrouhou, Slim Kallel, Mohamed Jmaiel. "A privacy-preserving traceability system for self-sovereign identity-based inter-organizational business processes", Computer Standards & Interfaces, 2025 Publication	4%
3	Puneeta Singh, Shrddha Sagar, Sofia Singh, Haya Mesfer Alshahrani, Masresha Getahun, Ben Othman Soufiene. "Blockchain-enabled verification of medical records using soul-bound tokens and cloud computing", Scientific Reports, 2024 Publication	1%
4	polygonscan.com Internet Source	<1%
5	www.stateregstoday.com Internet Source	<1%
6	etherscan.io Internet Source	<1%
7	eips.ethereum.org Internet Source	<1%
8	www.humanities.gov Internet Source	<1%
	ebin.pub	