UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT

SUBJECT: BLOCKCHAIN AND APPLICATION

TOPIC: BUILD A DECENTRALIZED BLOCKCHAIN WALLET WITH WEB3

LECTURER:

TRINH QUANG VIET, PHD

PHAN THANH GIANG K214162145

Ho Chi Minh City, January 1st, 2025

ACKNOWLEDGMENTS

First and foremost, I am deeply thankful to our advisor, Dr. Trinh Quang Viet, for his invaluable guidance, expertise, and encouragement throughout this journey. His insightful advice and constructive feedback have been instrumental in shaping this project.

I also extend our appreciation to the Faculty of Information Systems at the University of Economics and Law for providing me with the resources and knowledge that laid the foundation for work. Special thanks go to our classmates and peers for their collaboration and sharing of ideas, which enriched our understanding and approach.

Lastly, I am grateful to our families and friends for their unwavering support and understanding during this challenging but rewarding process. Their encouragement has been our source of strength and motivation.

Sincerely,

Ho Chi Minh City, January 1, 2025

Phan Thanh Giang

COMMITMENT

I hereby declare that this project is the result of my own research and efforts. All findings, analyzes and results presented in this report are original and have not been copied or reproduced from any other individual or group.

I take full responsibility for the accuracy and authenticity of the information provided in this project to ensure professionalism.

Sincerely,

Ho Chi Minh City, January 1, 2025

Phan Thanh Giang

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ừ	Mô tả
IS	Management Information Systems
.	…

1. Introduction

In recent years, blockchain technology has emerged as one of the most disruptive innovations in the fields of finance, technology, and beyond. With its decentralized, transparent, and secure architecture, blockchain has paved the way for groundbreaking applications, particularly in cryptocurrency and digital wallets. Among these, building a decentralized blockchain wallet using Web3 technology represents a significant step forward in enhancing the security and autonomy of financial transactions. This project focuses on implementing such a solution to provide a user-friendly and secure environment for managing digital assets.

1.1 Reasons for choosing topic

The rapid advancement of blockchain technology has created an undeniable shift in how individuals and organizations perceive financial systems. Traditional financial methods often face challenges such as centralization, lack of transparency, and vulnerability to cyberattacks. A decentralized blockchain wallet addresses these issues by leveraging blockchain's core principles of security, transparency, and decentralization.

Additionally, the increasing adoption of cryptocurrencies and decentralized finance (DeFi) platforms highlights the growing demand for tools that enable seamless interaction with blockchain networks. Recognizing this, we chose this topic to contribute to the development of innovative financial tools and provide users with greater control over their assets.

1.2. Urgency of the topic

The centralized nature of traditional banking systems exposes users to risks such as data commission and misuse of personal information. Moreover, the global shift toward digital currencies and decentralized applications (dApps) underscores the urgency to create solutions that align with these emerging trends.

A decentralized blockchain wallet not only empowers users by eliminating intermediaries but also ensures the integrity of their transactions through cryptographic security. This project addresses a timely and vital need, making it a significant contribution to the blockchain and Web3 ecosystem.

1.3 Objective of the topic

- Design and develop a decentralized blockchain wallet using Web3 technology that allows users to securely manage their digital assets. The wallet will enable users to perform essential operations such as creating accounts, sending and receiving cryptocurrency, and viewing transaction histories in a transparent and secure manner.

- Provide a user-friendly interface, ensuring accessibility for both novice and experienced users.

1.4 Scope of project

The scope of this project is focused on the development of a functional decentralized blockchain wallet using Web3 technologies.

This research is limited to Ethereum-based networks and smart contract interactions, with the aim of recreating practical applications of blockchain wallets. And it does not include the ability to scale and deploy wallets on public networks.

2. Theoretical Basis

2.1 Fundamental Concepts in Blockchain Technology

Blockchain is a distributed database that stores records of every transaction or digital event that has taken place and is shared between participants in the system. Transactions are verified by a majority of network members. This technology records transactions in a distributed digital ledger, which ensures immutability and cannot be tampered with. Any valuable asset, such as real estate, cars, or other types of assets, can be recorded as a transaction on the Blockchain.

Blockchain works without using a server or central system to store data. Instead, data is distributed across millions of computers around the world, all connected to the Blockchain network. This structure allows for public authentication of data because information is stored on every node in the network and can be easily verified.

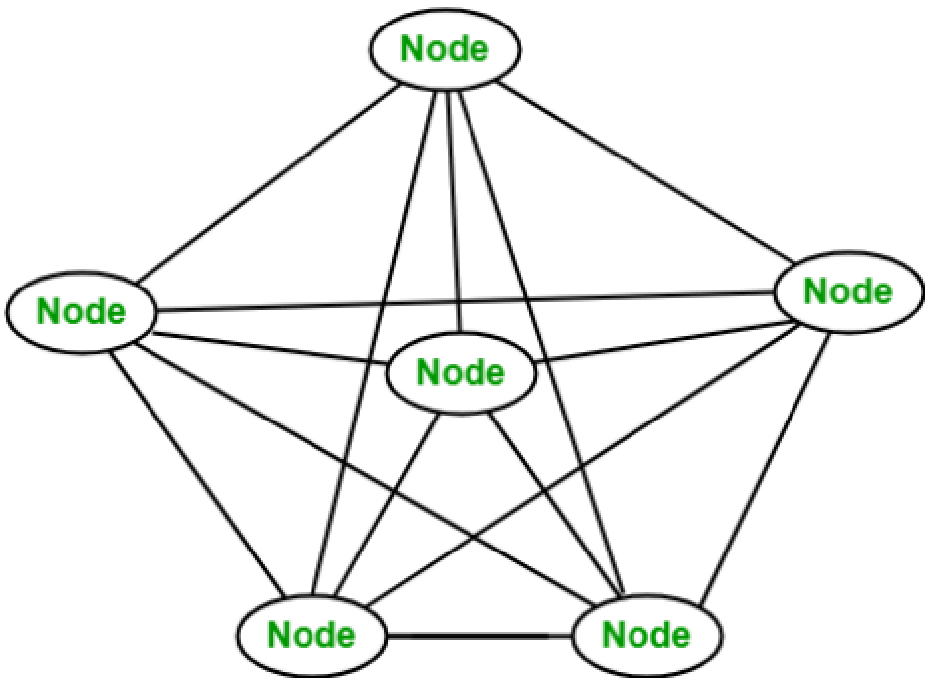

Figure: Fully Connected Peer-to-Peer Network. Sources: geeksforgeeks.org.

A node is a computer connected to the Blockchain network through a client software. This client software is responsible for verifying and transmitting transactions on the Blockchain. When a computer connects to the Blockchain, the system downloads a copy of the Blockchain data and synchronizes it with the latest data blocks in the network. Nodes participating in the Blockchain network also play the role of conducting transactions and receiving rewards, called "miners".

Benefits:

1. **Decentralization:** Blockchain technology is decentralized in nature, eliminating the need for intermediaries. This not only reduces costs but also increases transparency in transactions.
2. **Security:** Transactions on the blockchain are protected by advanced encryption algorithms, making them nearly impossible to hack or tamper with.
3. **Transparency:** Blockchain allows all parties involved to access the same transaction data, ensuring transparency and minimizing the risk of disputes.
4. **Efficiency:** Transactions on blockchain are processed quickly and efficiently, saving time and costs compared to traditional systems.
5. **Trust:** Blockchain's high transparency and security help build solid trust between transaction parties, creating the foundation for long-term relationships.

2.2. Web3 Principles and Frameworks

Web3 is the next generation of the internet, focusing on decentralization, transparency and user data ownership, developed on the blockchain platform. Unlike Web2, where data is mainly managed by centralized organizations, Web3 gives control of data and transactions directly to users through decentralized applications (Dapps). Principles such as **data source transparency**, **transaction latency management**, and **increased data readability** (e.g., hashes or smart contract addresses) are introduced to help users understand the ongoing operations and ensure data security.

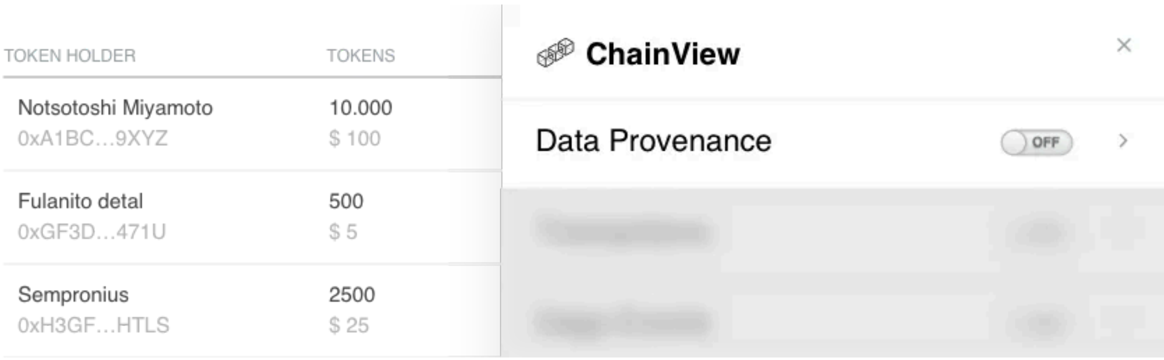

Figure: Design with css to clearly distinguish Blockchain data. Sources: medium.com

One of the strengths of Web3 is the transparency it brings. Principles such as **displaying transaction history, clearly indicating data origin** (blockchain or not), and **creating a dedicated view of blockchain data** (chain-view) are designed to ensure that users understand that all information and interactions in Dapp are trustworthy.

2.3. Smart Contract Standards

Smart contracts are an attractive feature of blockchain technology, which are pieces of code executed on the blockchain that facilitate, execute, and ensure the terms of an agreement between mutually untrusted parties without the need for their consent. Engage a trusted third party. Introduced by Szabo in 1994, the concept only really took off with the advent of blockchain. Smart contract standards help ensure compatibility and usability across various blockchain platforms like Ethereum. In addition, it is extremely important in ensuring flexibility, security and efficiency for decentralized applications.

Main Standards of Smart contracts:

Smart Contract Code

A piece of code that is stored, verified, and executed on the blockchain. These contracts operate independently based on the programming language and blockchain features used, allowing for the creation of complex rules and automation of transactions.

Smart Legal Contracts

Smart Legal Contracts is a code that complements or replaces traditional legal contracts. These contracts depend on legal and political institutions, rather than relying entirely on technology. This standard is suitable for applications requiring strict legal compliance

2.4. API Integration Principles

In today's rapidly evolving technology landscape, blockchain technology and APIs (Application Programming Interfaces) are transforming the way businesses handle data, transactions and digital interactions. API integration guidelines includes:

- **Standardized Communication**

Blockchain often uses complex protocols and data that is difficult to process directly, such as addresses, cryptographic signatures, or block data structures. Therefore, APIs act as a "bridge" between the application and the blockchain.

- **Simplifying Interaction with Blockchain**

APIs help simplify this process by converting data into familiar formats that are easier to read and process. This makes it easy for traditional applications to integrate blockchain without too many architectural changes.

- **Improve User Experience**

The API supports real-time features, such as transaction updates or status notifications when a new block is created. Applications can receive information as soon as the blockchain changes, helping users have a smoother and more accurate experience. This is especially useful for financial or transaction tracking applications.

- **Enhanced Security**

Security is indispensable in integrating APIs with blockchain. The API provides additional layers of security such as:

- Authentication: Use methods like API keys or OAuth to ensure only authorized parties have access.
- Encryption: Communication between the API and the application must be encrypted (HTTPS) to prevent data exfiltration attacks.
- Traffic control: Limit the number of API requests (rate limiting) to avoid denial of service (DDoS) attacks.

- **Scalability**

As the system grows, the API needs to be designed to easily scale and handle large numbers of requests. Techniques such as load balancing and API versioning help ensure that the API operates stably even when the number of users increases. The APIs will support multiple blockchains (multi-chain), allowing applications to interact with Ethereum

3. Related Work

3.1. Overview of Blockchain Technology

Recent research emphasizes blockchain's versatility in addressing modern challenges. For example, its application in healthcare ensures secure data sharing, while in finance, it accelerates cross-border payments and reduces transaction costs (Taherdoost, 2023). Furthermore, the integration of blockchain with emerging technologies like the Internet of Things (IoT) amplifies its potential to automate and secure decentralized networks (Alsharari, 2021).

Blockchain technology was originally developed to underpin Bitcoin. But gradually, the technology increasingly expanded its applications beyond cryptocurrencies. Now, blockchain is convenient for diverse use cases, including supply chain management, identity verification, and secure data exchange. Its core characteristics, such as decentralization, immutability, and transparency, eliminate the need for intermediaries and enhance trust among stakeholders (Habib et al., 2022).

Besides, blockchain faces limitations, such as scalability issues, high energy consumption, and the need for standardization. Ongoing advancements, including Layer-2 solutions and energy-efficient consensus mechanisms, aim to address these challenges, ensuring blockchain remains a transformative force across industries (Taherdoost, 2023).

3.2. Web3 and Blockchain Interoperability

According to Hosseini Bamakan & Banaeian Far, 2025, Web3 transitions from a centralized system to a decentralized network, allowing users to maintain control over their data. The absence of a central government reduces the risks associated with data breaches and enhances trust among participants. In addition, this group of authors also shows that Web3 allows the creation of applications in which users can verify the source and authenticity of data, which is important in financial and healthcare systems.

Besides web3, Smart contracts serve as the backbone of blockchain interoperability, enabling automated, trustless interactions across networks. For instance, Ethereum and its associated smart contract standards (e.g., ERC-20) provide a robust framework for creating interoperable decentralized applications. APIs will help connect traditional systems to the blockchain network, simplifying interactions for developers and end users. They provide a standardized interface to access blockchain functions, thus facilitating the widespread adoption of Web3 technology in industries such as finance, logistics and digital identity (Roldán Martínez, 2024).

3.3. Decentralized Blockchain Wallets

Decentralized wallets are essential tools in the blockchain ecosystem, enabling users to manage their digital assets and identities without reliance on intermediaries. These wallets leverage blockchain technology to enable users to interact with decentralized applications (dApps), execute transactions, and manage cryptographic keys.

Key Features	Functionality
Self-Custody	According to Nguyen et al. (2023), decentralized wallets such as MetaMask and Trust Wallet grant users full control over their private keys by storing them locally on user devices.
Cryptographic Security	The publication by Alharby and van Moorsel (2023) shows that decentralized wallets rely on cryptographic algorithms such as Elliptic Curve Digital Signature Algorithm (ECDSA) to secure private keys and transactions.
Seed Phrases and Key Management	Nguyen et al. (2023) emphasize the importance of deterministic key generation standards, such as BIP-39, used in decentralized wallets.
Compatibility with Multiple Blockchain Networks	According to Nguyen et al. (2023), many decentralized wallets support multi-chain functionality, enabling users

	to interact with multiple blockchain ecosystems, including Ethereum, Binance Smart Chain, and Polygon.
Integration with Decentralized Applications	Nguyen et al. (2023) note that decentralized wallets act as essential gateways to dApps, providing users with the ability to seamlessly sign in, execute transactions, and interact with smart contracts.

3.4. API Integration in Blockchain Wallets

APIs provide a standardized mechanism for connecting disparate systems, ensuring secure and efficient communication across platforms. APIs in blockchain wallets perform multiple functions, such as facilitating user authentication, transaction management, and real-time data retrieval. According to Hefny (2022), open APIs enhance the efficiency of e-payment systems by providing seamless integration with blockchain frameworks. This approach improves financial transaction processes, reduces costs, and fosters interoperability among financial systems. Blockchain wallets leverage APIs to:

1. **Access Blockchain Data:** APIs enable wallets to retrieve transaction histories, account balances, and real-time blockchain updates.
2. **Execute Transactions:** REST and WebSocket APIs are commonly used to initiate and confirm transactions, ensuring a smooth user experience.
3. **Enhance Security:** APIs allow for the implementation of authentication mechanisms such as OAuth or API keys to control access to blockchain functionalities.

Pasdar et al. (2021) highlight that blockchain oracles solve the “oracle problem” by securely transferring external data into the blockchain ecosystem, thereby expanding the functional scope of smart contracts, allowing Applying blockchain technology to the real world.

- **Benefits of API Integration**

- Improved Usability**:** APIs abstract the complexity of blockchain interactions, allowing developers to build user-friendly interfaces.
- Scalability: Modular API design facilitates scaling blockchain wallets to accommodate increasing user demand and transaction volumes.
- Real-Time Updates: APIs enable wallets to provide real-time notifications and updates, enhancing user engagement and trust.
- Cross-Platform Compatibility: By using APIs, blockchain wallets can interact with multiple blockchain networks and external systems, fostering interoperability.

- **Challenges and Considerations**

While APIs significantly enhance the functionality of blockchain wallets, their implementation poses challenges:

- Security Risks: Improper API implementation can expose wallets to vulnerabilities, such as unauthorized access or data breaches.
- Data Consistency: Ensuring synchronization between on-chain and off-chain data remains a technical challenge.
- Performance Optimization: Designing APIs to handle high transaction volumes without latency is critical for user satisfaction.

3. Data Description

3.1. Database Diagram

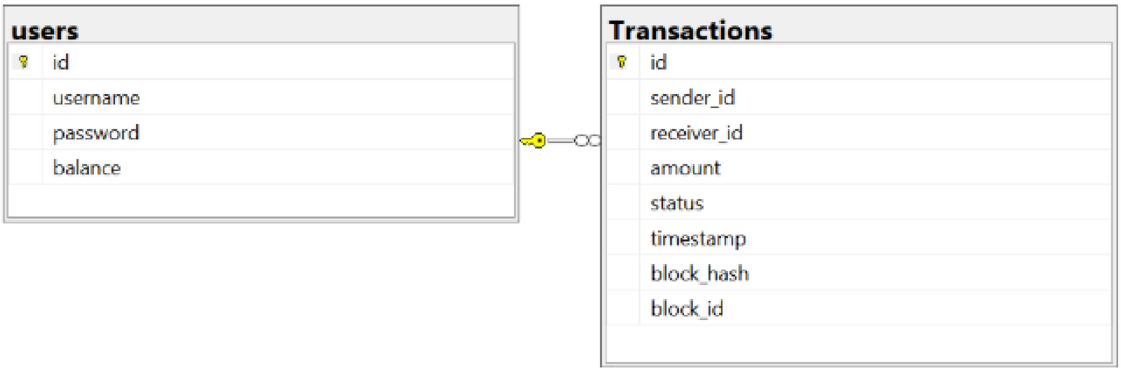

Figure: Blockchain Database Diagram. Source: Author.

Here, I have created three tables: users, transactions, and blocks and linked them together to form a comprehensive database that supports operations such as user management, transaction processing, and building a secure, transparent blockchain. This structure not only ensures operational efficiency but also provides a solid foundation if there is an opportunity to expand the system in the future.

3.2. Design dataBase Table

Users Table

Column Name	Data Type	Allow Nulls
id	int	☐
username	nvarchar(50)	☐
password	nvarchar(255)	☐
balance	float	☐
		☐

- id: Is the primary key column of the table, with data type **int**

- username: Column that stores the user's login name, with data type **nvarchar(50)**.
- password: Column stores encrypted passwords, with data type **nvarchar(255)**.
- balance: Column stores the user's account balance, with **float** data type.

Transactions Table

Column Name	Data Type	Allow Nulls
id	int	☐
sender_id	int	☐
receiver_id	int	☐
amount	float	☐
status	nvarchar(50)	☐
timestamp	datetime	☐
block_hash	nvarchar(255)	☐
block_id	int	☑
		☐

- id: Primary key column of the table, with data type **int**.
- sender_id: Stores the ID of the sender in the transaction.
- receiver_id: Stores the ID of the recipient in the transaction.
- amount: Stores the transaction amount, with **float** data type.
- status: The status of the transaction, with data type **nvarchar(50)**.
- timestamp:Time of the transaction, data type **datetime.**
- block_hash: The hash value of the transaction, data type **nvarchar(255).**
- block_id: ID of the block this transaction belongs to.

Blocks Table

Column Name	Data Type	Allow Nulls
	int	☐
[index]	int	☐
timestamp	datetime	☐
transactions	nvarchar(MAX)	☐
proof	bigint	☐
previous_hash	nvarchar(255)	☐
		☐

- id: Primary key column, **int** data type.
- index: The block index column, data type **int**.
- timestamp: Block creation time, with **datetime** data type.
- transactions: List of transactions in the block, with data type **nvarchar(MAX)**.

4. Methodology

4.1.Process flow diagram

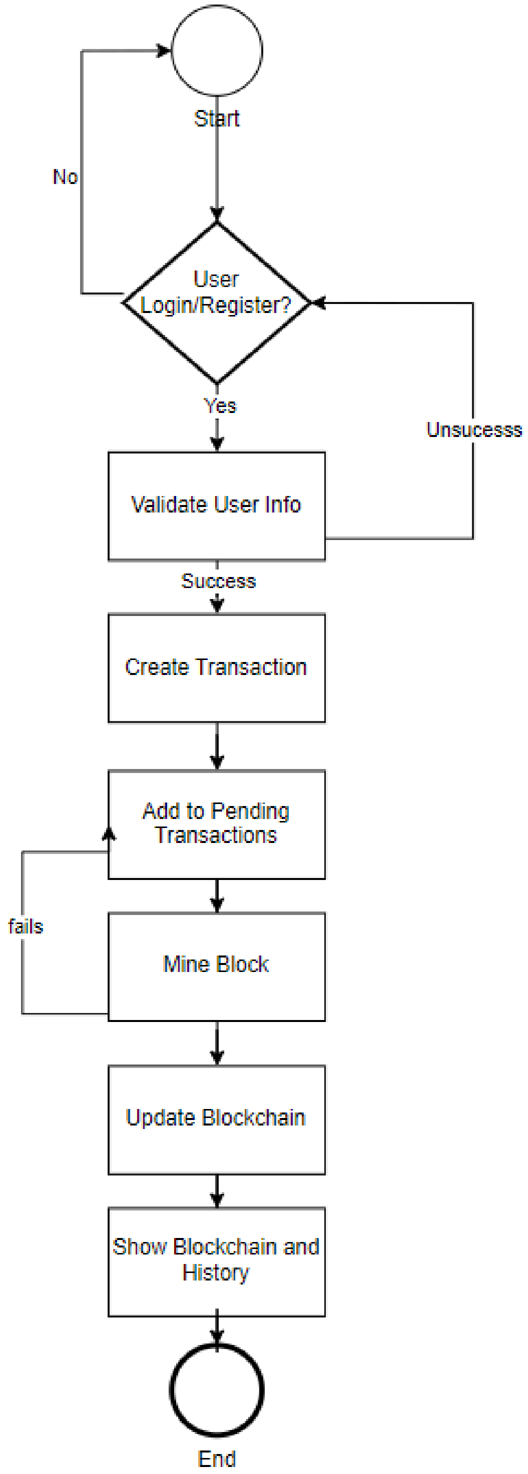

Figure: Blockchain-Based Transaction Flow Diagram. Sources: Author

First, the user logging into the system or registering for a new account. If the user fails to log in or provide valid registration details, the system prompts them to retry, maintaining security and preventing unauthorized access.

After login successfully, the system validates their information, such as their account balance and username. The blockchain plays a key role here by serving as a decentralized ledger that securely stores the user’s transaction history and balance.

Next, the user creates a transaction by specifying the recipient's username and the transaction amount. This transaction is added to the pending transactions pool, where it awaits processing. The blockchain becomes the backbone of this process as it ensures that every transaction is immutable, timestamped, and securely recorded. At this stage, the transaction is merely a proposal that requires further verification and inclusion in the chain.

After the transaction is created, it moves into the mining phase. Here, the system groups the pending transactions into a block. To validate this block, the system engages in a mining process, solving a computational puzzle to ensure the block’s legitimacy. This is where the blockchain’s proof-of-work mechanism shines—adding a layer of security and preventing malicious actors from tampering with the chain. If mining fails for any reason, the transaction remains in the pending state, safeguarding it from being lost or improperly processed.

Once the block is successfully mined, it is added to the blockchain. This step is the heart of the process, as the blockchain now incorporates the newly validated block, chaining it securely with all preceding blocks. Each block references the previous one via its hash, creating an unbreakable sequence of transactions.

Finally, the system provides feedback to the user by displaying the updated blockchain and the transaction history. The user can see their transaction securely embedded in the chain, demonstrating how the blockchain ensures the reliability and integrity of the system.

4.2. Frontend

The frontend interface in our Blockchain project is built with HTML, CSS, in which Web3.js plays a central role in connecting and communicating with the blockchain.

Aim: Create an intuitive, easy-to-use user experience, while integrating complex blockchain features efficiently.

4.3. BackEnd

- Using Python, leveraging the Flask framework for building web applications. The primary objective of the backend is to manage blockchain functionalities, including user authentication, transaction processing, and blockchain maintenance, while providing robust communication between the frontend and the underlying blockchain structure.

- The backend is structured into distinct components, each responsible for specific functionalities. Flask's blueprint feature organizes these components, with files like auth.py and transactions.py handling user authentication and transaction management respectively. The authentication module ensures secure user registration, login, and session management, integrating features like password hashing and session tokens to enhance security.

- For transaction handling, the backend supports the creation, validation, and addition of transactions to a pending pool. This is followed by mining new blocks that encapsulate these transactions, ensuring the integrity of the blockchain. Once mined, the block is appended to the blockchain and relevant balances are updated in the database.

- SQLAlchemy integrated with Flask, which is utilized for ORM (Object-Relational Mapping), enabling seamless interaction with the Microsoft SQL Server database. The database itself stores essential entities such as Users, Transactions, and Blocks, ensuring data persistence and integrity across the blockchain system.

4.4. Database

In this project, the database is used to store and manage information related to users, transactions, and blocks in the blockchain. The system will use data from SQL Server to ensure consistency and effective data querying. Includes 3 panels:

- Users table to store user information such as username, encrypted password, and balance;
- Transactions table to manage transaction details such as sender, recipient, amount, status, and link to the block containing the transaction.
- The blocks table, which stores blocks in the blockchain with information about their position in the chain, a list of transactions, and the hash of the previous block. When a user makes a transaction, data is added to the transactions table and then processed into blocks in the blocks table.

5. Results

5. 1. Design Interface

5.1.1. Login Page

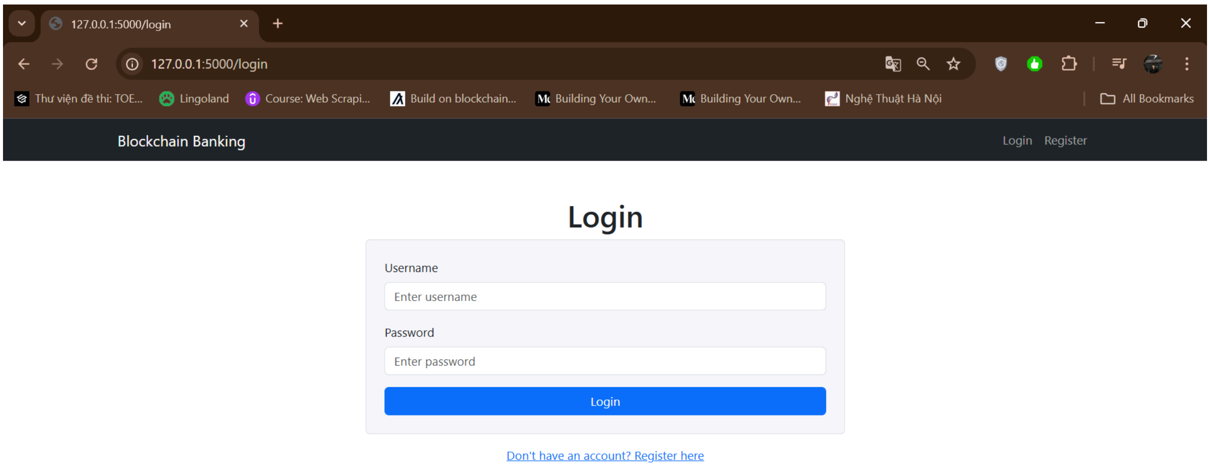

Figure: Login Page. Sources: Author.

Before trading, users are required to log in with a username and password, to access their account. For new users, the link to the registration page is provided at the bottom.

5.1.2. Sign-up Page

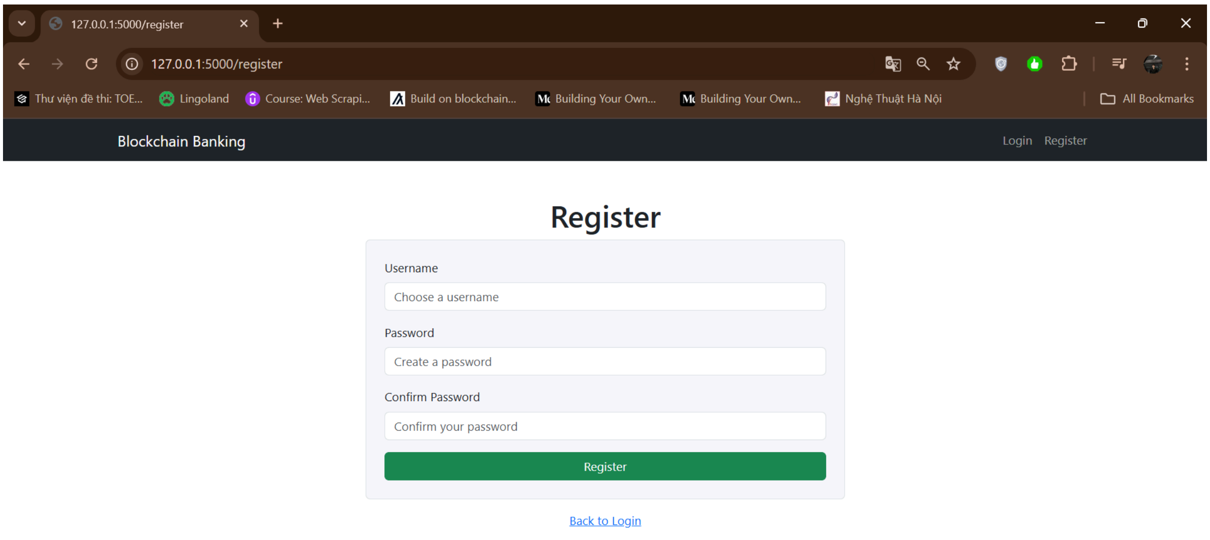

Figure: Sign-up. Sources: Author.

When users do not have an account and cannot log in to the platform, they can click the create registration page to create a new account. Registering a new account requires users to create a unique username, set a secure password, and confirm its accuracy. After successful registration, the user is redirected to the login page to access the system.

5.1.3. Transaction Page

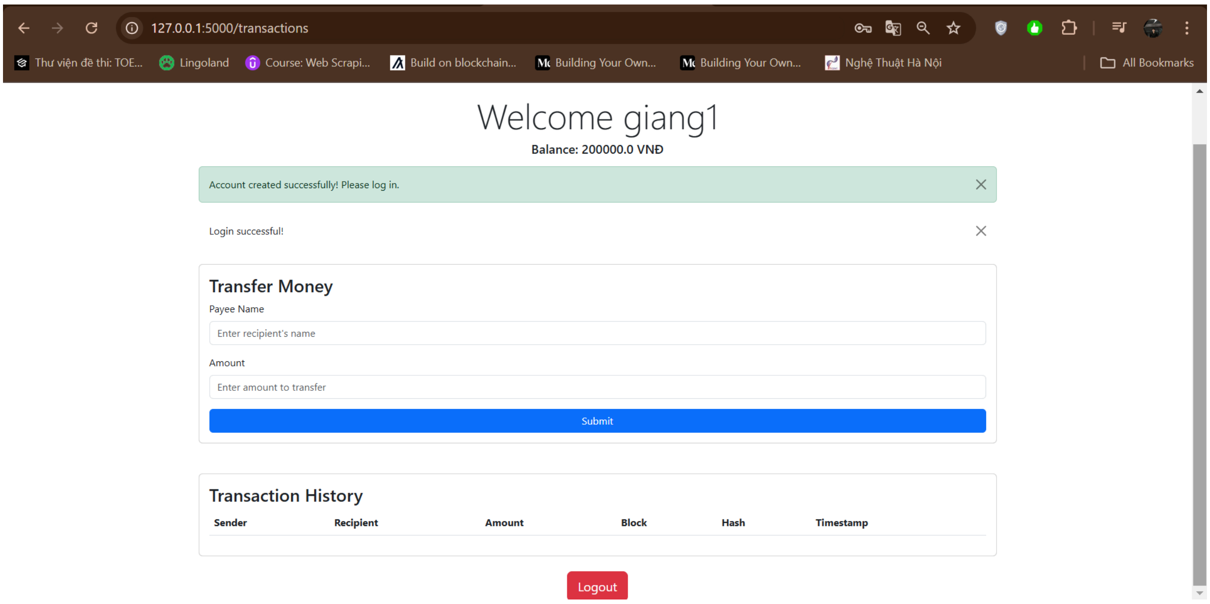

Figure: Transaction Page. Sources: Author.

After successful login, the user is redirected to the transaction page. Here, users can make money transfers, view transaction history, and check account status. The intuitive interface displays account name information, current balance, and input fields such as recipient and amount to transfer.

5.2. Overview of Features

5.2.1. Login and Registration Pages

The login page can be viewed as a HomePage and linked to the registration page. Users can register for a new account by providing basic information such as username and password, and confirm the password. After successful registration, users can log in to perform other functions such as creating transactions, viewing transaction history or mining new blocks.

5.2.2. Transaction

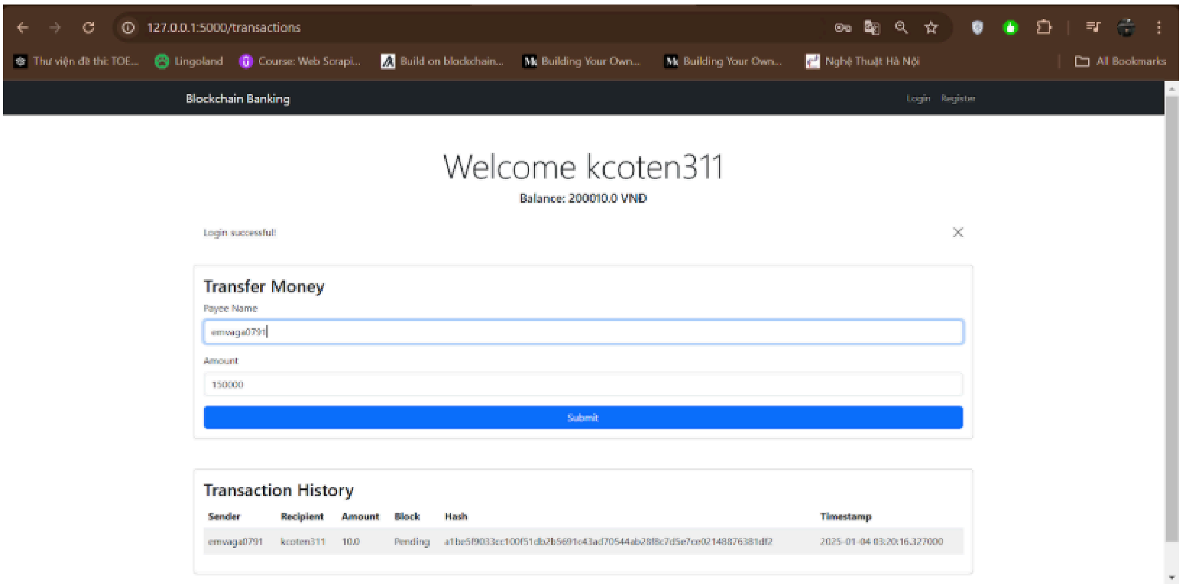

Figure: Transaction Process. Sources: Author.

When a new user registers an account, the system will default to initializing an account balance of 200,000 VND. On the transaction page, users can make a money transfer by entering Payee Name (recipient name) and the amount to transfer (Amount) in the corresponding fields. After pressing the Submit button, the system will check the information and perform the transaction if the conditions are met, such as the account balance must be sufficient to make the transaction.

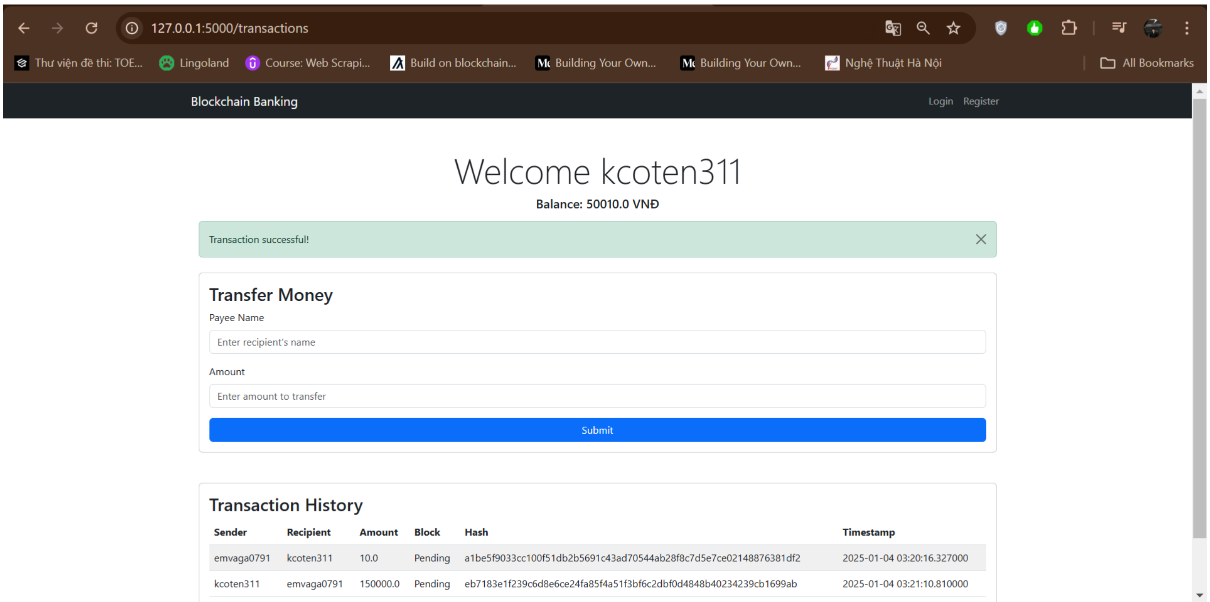

Figure: Successful transaction on the sender's side. Sources: Author.

For example: I enter Payee Name as ‘emvaga0791’ and the amount to transfer is 150,000 VND. After a successful transaction, my account only has 50010.0 VND left and it will display the green message 'Transaction successful!'.

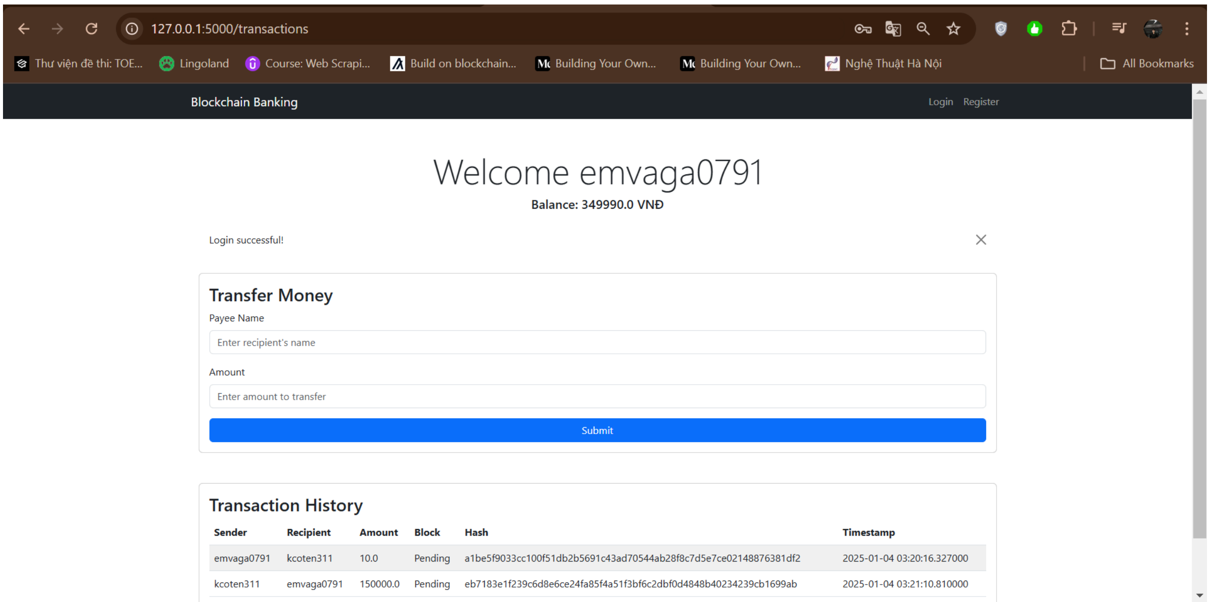

Figure: Transaction successfully on the recipient's side. Sources: Author.

After user kcoten311 transfers 150,000 VND to emvaga0791, the blockchain will process the transaction by validating the input and updating the respective balances of both accounts. It can be seen that the kcoten311 balance decreased by 150,000 VND, leaving 50,010 VND. On the contrary, the recipient account emvaga0791 reflected that the transferred amount had been successfully received, increasing the balance to 349,990 VND.

This transaction is recorded in the blockchain system as a pending transaction and included in the Transaction History, demonstrating transparency. Additionally, the system displays a confirmation message such as “Transaction successful!” for senders, ensure users are promptly notified of their transaction status.

=> This mechanism highlights blockchain's ability to manage and reflect financial transactions securely and accurately.

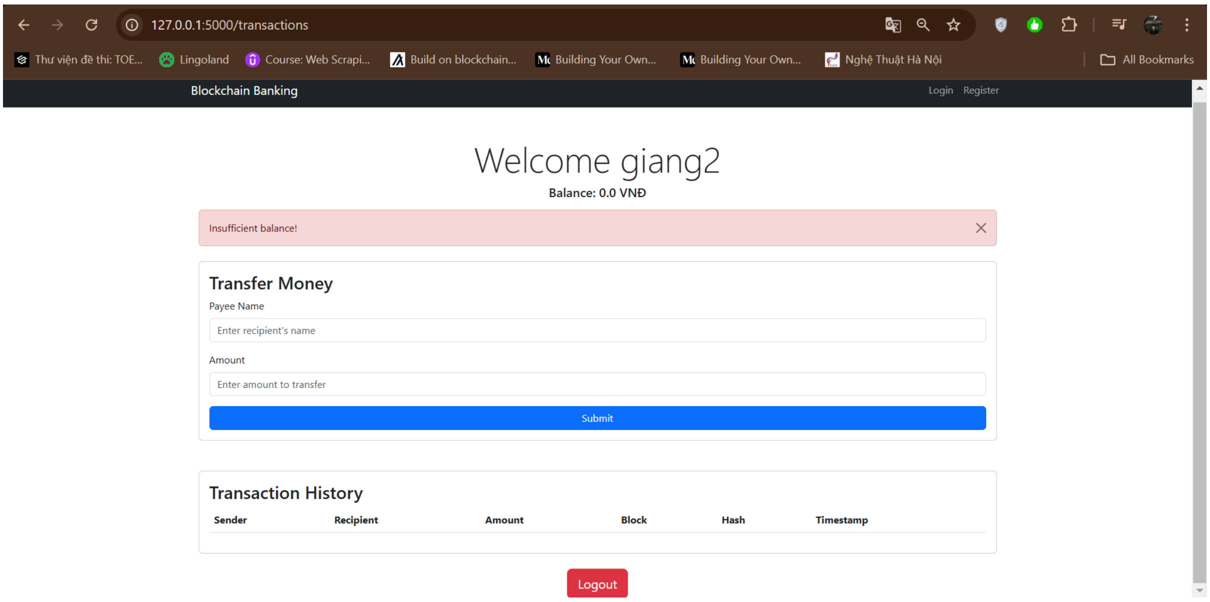

Figure: Transaction failed. Sources: Author.

In case the user tries to make a transaction when the account balance is insufficient, the system will display a clear message saying "Balance is not enough!" at the top of the interface view. This helps users to be aware of currently unavailable transactions and avoid confusion.

5.2.3. Transaction History

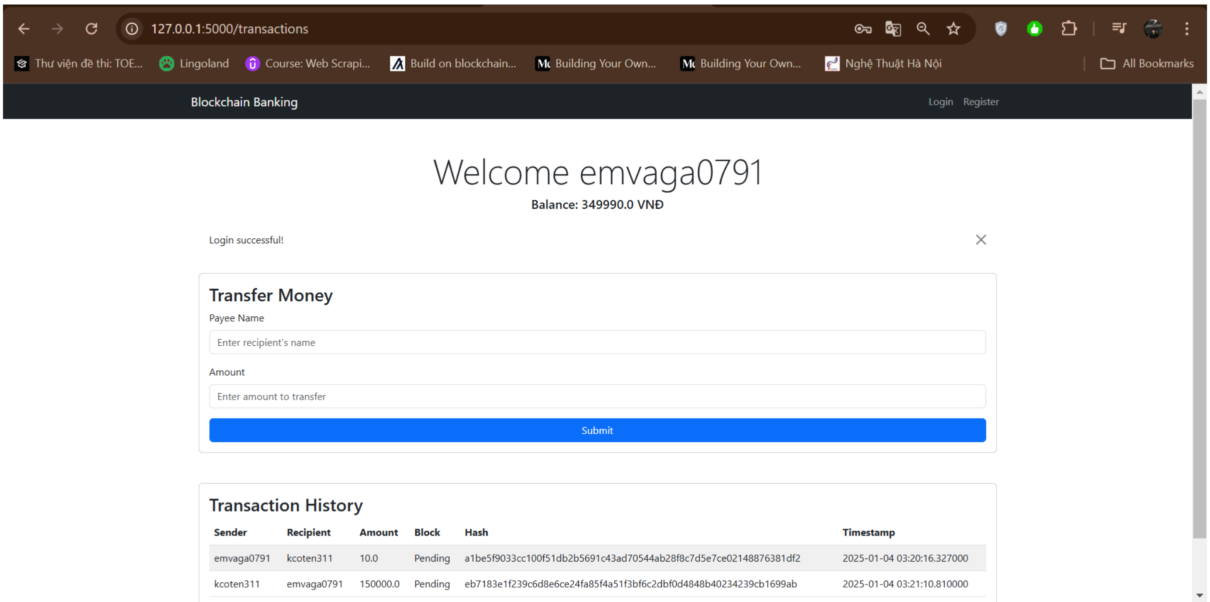

Figure: Transaction History. Sources: Author.

Transaction information will be stored and displayed in the list below, including: sender, recipient, amount, block containing the transaction, hash code of the transaction, and execution time.

5.2.4. Save Database

	id	sender_id	receiver_id	amount	status	timestamp	block_hash	block_id
1	1	1	2	10	Pending	2025-01-04 03:20:16.327	a1be5f9033cc100f51db2b5691c43ad70544ab28f8c7d5e7ce...	NULL
2	2	2	1	150000	Pending	2025-01-04 03:21:10.810	eb7183e1f239c6d8e6ce24fa85f4a51f3bf6c2dbf0d4848b402...	NULL
3	3	2	1	10	Pending	2025-01-04 03:22:19.737	4c1debab8704837b072f43ca4d15843dbdfc1aac20a4dfa755...	NULL
4	5	1	2	50000	Pending	2025-01-04 03:32:53.903	aa6885c8b3360e0796dd7d3a8aee0689d505d8be9e88d341...	NULL
5	6	1	2	50000	Pending	2025-01-04 03:32:59.237	25deab12cf9fd4d2800e7e9583e1f49dce42677f7fc9d9a14b7...	NULL
6	7	1	2	50000	Pending	2025-01-04 03:33:08.540	087f780e75dff9eebe8ee0e91e96c706f648e680bf9c2924879...	NULL
7	8	1	2	50000	Pending	2025-01-04 04:16:41.683	28498cf0cd0057a1ba5da137f3a5f2bca233d52566f0d4a609...	NULL
8	9	1	2	50000	Pending	2025-01-04 04:16:49.627	f90e89a45f3c4776d0fe3cd3ce566ccc4f51e2c70bb4d97d470...	NULL

Figure: Save Data. Sources: Author

This feature is intended to maintain data integrity, providing users and administrators with an organized and auditable record of every transaction made on the platform. By systematically saving transaction data, the system enables historical tracking, authentication, and verification of all blockchain-related activities.

In this case, when saving the data for each transaction. Some attributes such as sender_id, receiver_id, amount, status, timestamp, block_hash, and block_id. For instance, a transaction initiated by user 1 sender_id to user 2 receiver_id for an amount of 10 VNĐ is logged with a status of Pending, a unique block_hash to ensure cryptographic integrity, and a timestamp for chronological ordering. If this transaction is successfully included in a mined block, the corresponding block_id will reflect its inclusion.

6. Discussion

6.1 Limitation

6.2 Expand the project in the future

7. Implications

Conclusion

References

[1] GeeksforGeeks. (n.d.). Blockchain Technology Introduction. Retrieved from https://www.geeksforgeeks.org/blockchain-technology-introduction/

[2] Beltran, B. (2018, March 30). Web3 Design Principles. Medium. Retrieved from https://medium.com/@lyricalpolymath/web3-design-principles-f21db2f240c1

[3] Alharby, M., & van Moorsel, A. (2017). *Blockchain-based smart contracts: A systematic mapping study*. School of Computing Science, Newcastle University, UK, & College of Computer Science and Engineering, Taibah University, KSA. Retrieved from https://arxiv.org/abs/1710.06372

[4] Roldán Martínez, D. (2024, December 10). *Blockchain API integration: Transforming systems with APIs - Leveraging blockchain and APIs for real-world applications*. API Conference Blog. Retrieved from https://apiconference.net/blog-en/blockchain-api-integration-guide/

[5] Hefny, M. H. M. (2022). Instant payment framework based on blockchain technology and open API architecture to improve the e-payment system of the local banks in Egypt. *Journal of Financial and Commercial Studies*, 32(3), 1048-1070. Retrieved from https://mosj.journals.ekb.eg/article_273409_2b51a4b1e7c46f1d0221174d325cc94c.pdf.

[6] Pasdar, A., Lee, Y. C., & Dong, Z. (2021). Connect API with Blockchain: A Survey on Blockchain Oracle Implementation. *ACM Computing Surveys*. https://dl.acm.org/doi/abs/10.1145/3567582.

[7] Kersic, V., Vidovic, U., Vrecko, A., Domajnko, M., & Turkanovic, M. (2023). Orchestrating digital wallets for on-and off-chain decentralized identity management. IEEE access.

[8] Raval, S. (2016). Decentralized applications: harnessing Bitcoin's blockchain technology. " O'Reilly Media, Inc.".

[9] Habib, G., Sharma, S., Ibrahim, S., Ahmad, I., Qureshi, S., & Ishfaq, M. (2022). Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. *Future Internet, 14*(11), 341. https://doi.org/10.3390/fi14110341

[10] Taherdoost, H. (2023). Smart Contracts in Blockchain Technology: A Critical Review. *Information, 14*(2), 117. https://doi.org/10.3390/info14020117

[11] Alsharari, N. (2021). Integrating Blockchain Technology with the Internet of Things for Efficiency. *International Journal of Technology, Innovation and Management, 1*(S1). https://doi.org/10.54489/ijtim.v1i2.25

[12] Hosseini Bamakan, S. M., & Banaeian Far, S. (2025). Distributed and trustworthy digital twin platform based on blockchain and Web3 technologies. *Cyber Security and Applications, 3,* 100064. DOI: 10.1016/j.csa.2024.100064

[13] Roldán Martínez, D. (2024). Blockchain API Integration: Transforming Systems with APIs. *API Conference Blog*. Retrieved from https://apiconference.net/blog-en/blockchain-api-integration-guide/