Crypto Wallet: An Educational Simulator for Cryptocurrency Transactions and Wallet Management

Mohammad Meezan [1], Mohammed Mehraj Pasha [2], Mustqeem Sannakki [3], Raheel Jawed [4], Prof. Deepika Dash*

*Assistant Professor, Computer Science and Engineering, R V College of Engineering

[1-4]BE students, Department of Computer Science and Engineering, R V College of Engineering

mohammadmeezan.in@gmail.com, Corresponding Author.

Abstract: *This paper introduces a single-page web-based simulator specifically developed for academic purposes to demonstrate the operations of cryptocurrency wallets and in-memory exchange trading. The system replicates essential cryptocurrency workflows, including wallet creation, encryption, asset transactions, and real-time INR valuations utilizing CoinGecko APIs. It illustrates contemporary cryptographic principles through the use of technologies such as BIP39, AES256-GCM, PBKDF2, and ECDSA. This platform provides a secure, self-contained environment conducive to learning blockchain concepts without the necessity of deploying an actual blockchain*

Keywords: *Cryptocurrency, Crypto Wallet, Web Simulation, BIP39, AES Encryption, ECDSA, CoinGecko API, React, Node.js, Fintech Education*

1.INTRODUCTION

Over the last decade, cryptocurrencies have seen a remarkable increase in popularity, leading to a significant interest in decentralized finance. This growing interest highlights the importance for students and learners to gain practical experience with cryptocurrency wallets, encryption, and trading platforms. However, many existing educational resources are often complex or require the use of actual cryptocurrencies, which can deter learners. To address this gap, there is a clear need for user-friendly educational tools that can effectively illustrate the real-time functions of cryptocurrency wallets and exchanges without exposing users to financial risks.

This paper introduces a web-based simulator designed for academic use, aiming to provide a simplified yet technically robust platform that emulates authentic wallet operations and trading mechanisms. The simulator replicates essential cryptocurrency workflows, including wallet creation, encryption, and asset transactions, by incorporating contemporary cryptographic principles such as BIP39, AES256-GCM, PBKDF2, and ECDSA. Additionally, it provides real-time INR valuations using CoinGecko APIs.

The platform offers a secure, self-contained environment, making it conducive for learning blockchain concepts without the necessity of deploying an actual blockchain. Unlike other tools that may lack customizability or require actual cryptocurrency, this simulator leverages open-source tools and APIs to create a comprehensive educational framework, providing features such as real-time INR valuation, multi-wallet management, and secure handling of encrypted mnemonics.

2. RELATED WORK

Cryptocurrency wallets have evolved significantly, prompting researchers to investigate authentication mechanisms, vulnerabilities, and secure implementation techniques. Some studies have classified wallet security based on the types of authentication factors used, which helps guide the development of robust security models in future tools [1]. Additionally, a taxonomy of vulnerabilities and design flaws in wallet interfaces has been explored, suggesting that well-structured UI design plays a critical role in preventing user-side security failures [2].

Advancements in cryptographic infrastructure, particularly with Trusted Execution Environments (TEEs), have opened new possibilities for wallet applications to offer both security and performance [3]. These technologies have been used to securely store private keys and perform sensitive operations within isolated environments, which directly supports the cryptographic goals of wallet simulators. Reviews of blockchain simulators have also revealed a lack of academic tools tailored for education and highlighted the importance of building frameworks that emphasize usability in a safe learning context [4].

Simulation models of cryptocurrency networks have further been developed to represent decentralized systems, offering insights into blockchain operations at a network level [5]. Additionally, surveys on blockchain simulators for IoT environments have emphasized the value of modular architectures that can be adapted across use cases [6]. The integration of such architectures into learning platforms could greatly enhance user interactivity and system flexibility.

To increase user engagement, gamified approaches to cryptocurrency education have been explored. These methods have demonstrated that users gain better conceptual understanding when abstract financial topics are presented through interactive simulations [7]. Gamification not only improves motivation and retention but also allows learners to experiment with decision-making in a risk-free environment, reinforcing economic and cryptographic concepts through experiential learning. By simulating real-world scenarios such as trading, wallet management, and investment strategies, gamified tools can bridge the gap between theoretical knowledge and practical application.

Furthermore, detailed educational guides have been proposed to assist in the development of classroom-focused cryptocurrency platforms [8]. These guides offer technical insights into tool design, such as incorporating encryption standards, API integration, and simplified user flows for beginners. They also emphasize the importance of aligning technical content with curriculum goals, ensuring that students not only interact with the system but also understand the underlying mechanisms. Together, these contributions underline the importance of educational simulators that combine pedagogical structure with engaging features to facilitate effective learning of decentralized technologies.

3. OBJECTIVES

1. Implement Stateless Authentication
One of the core goals of this project is to eliminate the reliance on server-side session storage by adopting a **stateless authentication** approach. This is accomplished through the use of **self-contained JWTs**, which encapsulate all necessary user information within the token itself. This design significantly reduces server memory consumption, allows for easier horizontal scaling, and improves system responsiveness. By removing the dependency on centralized session stores, the system becomes more suitable for **microservices**, **cloud environments**, and **highly distributed systems**.

2. Enhance Security Using Signed Tokens
Security is a top priority in any authentication system. This project aims to ensure strong protection against common threats such as **session hijacking**, **man-in-the-middle attacks**, and **token tampering** by leveraging **cryptographically signed JSON Web Tokens**. JWTs will be generated using a **secure secret key or public/private key pair** (depending on the signing algorithm used, such as HS256 or RS256), ensuring that the tokens cannot be altered or forged without detection. In addition, **token expiration**, **refresh tokens**, and **secure storage practices** will be incorporated to further safeguard the authentication process.

3. Enable Role-Based Access Control (RBAC)
A key functional requirement of this project is to implement **Role-Based Access Control (RBAC)** using JWTs. This involves embedding user roles (such as **Student**, **Teacher**, **Principal**, or **Administrator**) directly within the token payload. By doing so, the system can enforce **granular access policies**, allowing or restricting access to specific resources or actions based on the user's role. This ensures that each user interacts with the system according to their privileges, thereby enhancing both security and usability.

4. Improve Scalability and Cross-Platform Compatibility
To ensure the system's relevance and adaptability across various use cases, the project aims to deliver a solution that is both **highly scalable** and **platform-agnostic**. JWTs, being **compact and JSON-based**, are ideal for transmission across different platforms and devices—including **web applications**, **mobile clients**, and **IoT devices**. The stateless nature of JWTs also facilitates seamless integration with **load balancers**, **CDNs**, and **cloud-native services**, making the system suitable for handling large-scale user bases with high concurrency and performance demands.

4. METHODOLOGY

4.1. Frontend (User Interface):

- Technologies: The frontend is built using React.js, a popular JavaScript library for building user interfaces. It is enhanced with Tailwind CSS, which provides a streamlined and responsive design.
- Functionality: This part of the system provides the user interface for secure wallet management. It displays real-time transaction logs and allows users to interact with the simulator's features. The design is in light mode, offering a clean and intuitive experience, with transaction records presented in a tabular format and trading actions illustrated with smooth animated transitions.

4.2. Backend (Server-Side Logic):

- Technologies: The backend utilizes Node.js in conjunction with Express.js.
- Functionality: It establishes a RESTful API that governs essential functionalities such as wallet creation, encryption, and transaction simulation. This is where the core logic for processing requests from the frontend and interacting with the database and external APIs resides.

4.3. Database (Data Storage):

- Technology: MongoDB Atlas, a cloud-based database, is used for data storage.
- Functionality: It stores user wallets, encrypted mnemonic phrases, and transaction histories. This provides persistent storage, unlike conventional simulators that might rely solely on in-memory storage.

4.4. Third-Party API (Real-time Data):

- Technology: The CoinGecko API is used.
- Functionality: It retrieves real-time cryptocurrency prices, which refresh every thirty seconds to mimic a live market environment. This allows the trading simulation to use real-time exchange rates for cryptocurrencies like Bitcoin (BTC), Ethereum (ETH), Polygon (MATIC), Binance Coin (BNB), Litecoin (LTC), Dogecoin (DOGE), and Solana (SOL).

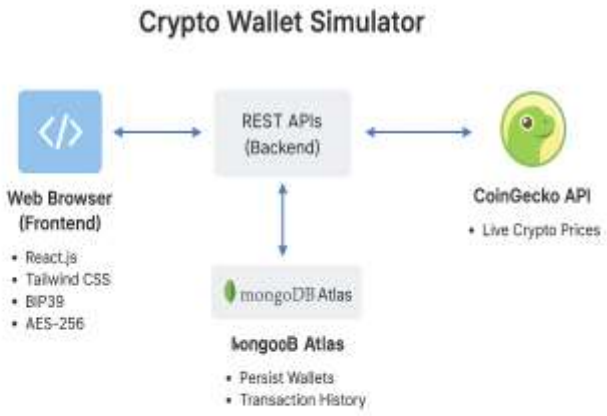
4.5. Cryptographic Methods (Security):

- BIP39: Used for the generation of secure 12-word mnemonic phrases.
- AES-256 in CBC mode: Employed for the encryption of wallet data and the generated mnemonic phrases.
- PBKDF2 (Password-Based Key Derivation Function 2): Used for deriving encryption keys from user passwords, ensuring a high level of cryptographic security.
- SHA-256 (Secure Hash Algorithm 256): Used for hashing transaction records, contributing to the immutability of these records.

4.6. User Features and Functionality:

- Wallet Creation: Users can create wallets through the application of BIP39-compliant 12-word mnemonic phrases.
- Multiple Wallet Management: Each user is permitted to create and manage multiple wallets within a single interface. Each wallet maintains its own encrypted mnemonic and independent transaction history, aiding in understanding wallet segregation and portfolio management.
- Trading Simulation: This encompasses the conversion of INR to supported cryptocurrencies and vice versa. Each transaction is timestamped, logged in a secure ledger, and presented in real-time on the user interface for review.

- Encryption and Security: Wallet data, including mnemonic phrases, is securely encrypted using AES-256 in CBC mode, with the encryption key generated via PBKDF2 from the user's password. This amalgamation of technologies guarantees that user data remains both secure and immutable while upholding system efficiency and responsiveness.



5.FEATURES

5.1 Wallet Creation and Security

The Crypto Wallet Simulator allows users to create wallets using BIP39-compliant 12-word mnemonic phrases. These mnemonic phrases are securely encrypted with AES-256 in CBC mode. A high level of cryptographic security is ensured as the encryption key is generated from a user-provided password via PBKDF2.

5.2 Trading Functionalities and Supported Cryptocurrencies

Users can simulate the purchase and sale of various cryptocurrencies using Indian Rupees (INR). These simulated transactions utilize real-time exchange rates obtained from the CoinGecko API. The simulator currently supports widely used cryptocurrencies, including Bitcoin (BTC), Ethereum (ETH), Polygon (MATIC), Binance Coin (BNB), Litecoin (LTC), Dogecoin (DOGE), and Solana (SOL).

5.3 Transaction Management and History

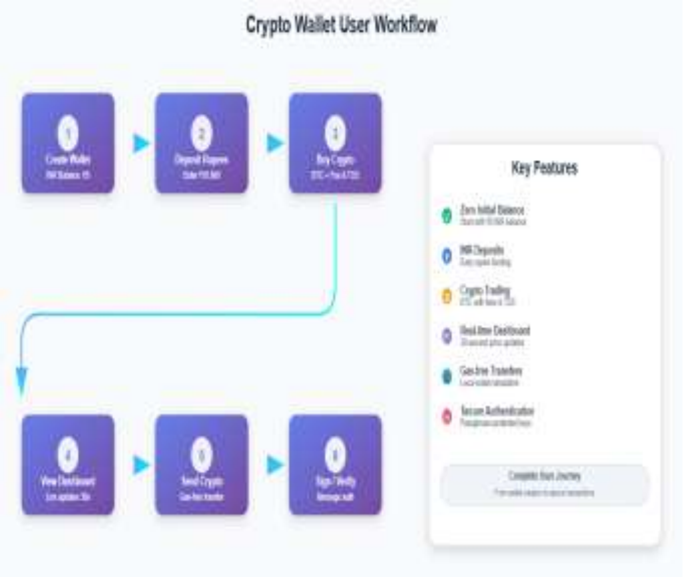
Every transaction is timestamped and documented in a secure ledger. Users can easily access and review this transaction history at their convenience. Each wallet maintains its own independent transaction history.

5.4 Multi-Wallet Management

Users have the capability to create and manage multiple wallets within a single interface. Each wallet retains its own encrypted mnemonic, which helps in understanding wallet segregation and managing multiple portfolios.

5.5User Interface and Experience

The user interface is designed in light mode, providing a clean and intuitive experience. Transaction records are presented in a tabular format with clear delineation. Trading actions are illustrated through smooth animated transitions, enhancing the educational and interactive quality of the application. These integrated features ensure that the simulator provides a comprehensive and realistic approximation of real-world cryptocurrency wallet operations, without the risks associated with actual financial transactions.

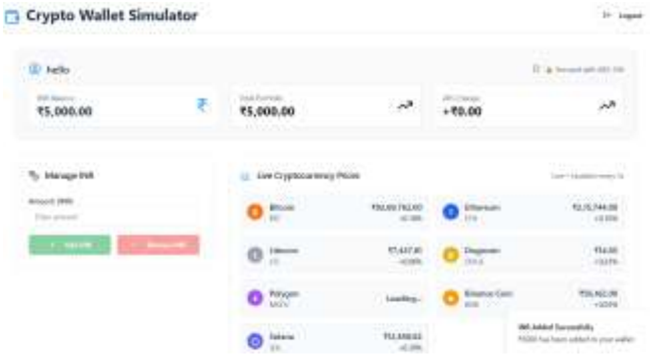


6.RESULTS

Figure 1. Crypto Wallet User Workflow

1. Core Functionality and Realism

The Crypto Wallet Simulator has been developed to proficiently illustrate the fundamental processes of cryptocurrency wallet creation, data encryption, and simulated trading. A key aspect of its realism is the integration of real-time Indian Rupee (INR) pricing data, which is continuously fetched from the CoinGecko API. This real-time data allows the simulator to mimic live market conditions, providing users with a dynamic and authentic experience of cryptocurrency value fluctuations during simulated trades. The successful implementation of these core functionalities ensures that the simulator serves as an effective educational tool, bridging the gap between theoretical knowledge and practical application by approximating real-world cryptocurrency operations.



Crypto Wallet Simulator

hello

₹5,000.00

₹5,000.00

+₹0.00

Manage INR

Live Cryptocurrency Prices

Bitcoin

Ethereum

Litecoin

Dogecoin

Polygon Loading...

Binance Coin

Solana

INR Added Successfully

2. Data Storage and Persistence

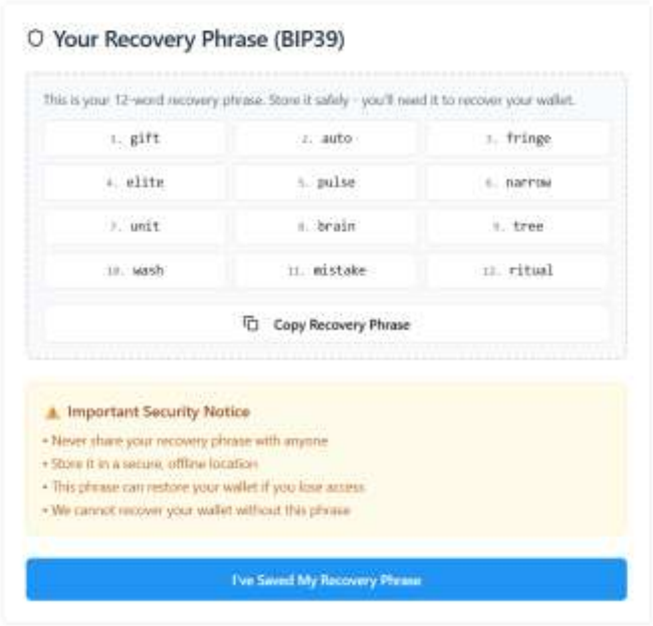
A significant advantage of this simulator over many conventional educational tools is its implementation with persistent storage through MongoDB. Unlike solutions that rely solely on in-memory storage, which means data is lost upon session termination, MongoDB ensures that all essential user data, including created wallets, encrypted mnemonic phrases, and detailed transaction histories, are permanently saved. This persistence is crucial for a meaningful learning experience, allowing users to track their progress, review past transactions, and manage multiple wallets over time, enhancing the simulator's utility for long-term educational engagement.

3. System Integrity and Performance

The application upholds a high standard of both data integrity and system performance through the rigorous application of contemporary cryptographic protections. Wallet creation leverages BIP39 for standardized mnemonic phrase generation, while AES-256 in CBC mode is used for robust encryption of sensitive wallet data. PBKDF2 ensures that encryption keys are securely derived from user passwords, adding another layer of security. Furthermore, SHA-256 is employed for hashing transaction records, contributing to their immutability and verifiable integrity. This comprehensive suite of cryptographic methods not only secures user data against unauthorized access and tampering but also ensures that the system operates efficiently and remains responsive, even under simulated trading activities.

4. Limitations

Despite its robust features and educational utility, it is important to acknowledge a fundamental limitation of the Crypto Wallet Simulator. As a simulation environment, it does not directly engage with a live blockchain network. Consequently, it does not include simulations of genuine blockchain-specific characteristics such as decentralization or the calculation and payment of gas fees. While this design choice allows for a risk-free and controlled learning environment, it means that aspects like network consensus, mining/staking rewards, or the intricacies of transaction finality on a distributed ledger are not replicated. This distinction is crucial for users to understand that the simulator provides a simplified model focusing on wallet and trading mechanics, rather than a full-fledged blockchain ecosystem.



Your Recovery Phrase (BIP39)

This is your 12-word recovery phrase. Store it safely - you'll need it to recover your wallet.

1. gift	2. auto	3. fringe
4. elite	5. pulse	6. narrow
7. unit	8. brain	9. tree
10. wash	11. mistake	12. ritual

Important Security Notice

- Never share your recovery phrase with anyone
- Store it in a secure, offline location
- This phrase can restore your wallet if you lose access
- We cannot recover your wallet without this phrase

6.CONCLUSION

The Crypto Wallet Simulator stands as a significant contribution to FinTech education, offering a user-friendly and secure platform designed to demystify cryptocurrency wallet creation, encryption, and trading. By providing a safe, sandbox environment, the simulator effectively bridges the gap between theoretical knowledge and practical experience. This allows students and learners to interact with complex cryptocurrency concepts without the inherent financial risks associated with real-world transactions. Its intuitive interface and clear presentation of processes make it an invaluable resource for anyone seeking to understand the foundational aspects of digital asset management.

A core strength of the simulator lies in its meticulous adherence to contemporary cryptographic principles. The implementation of BIP39 for mnemonic phrase generation, AES-256 for robust encryption, and PBKDF2 for secure key derivation ensures that the simulated environment closely mirrors the security standards of actual cryptocurrency wallets. This focus on cryptographic integrity not only educates users on the importance of secure practices but also builds trust in the simulated operations, making the learning experience more authentic and impactful.

Furthermore, the integration of real-time market data through the CoinGecko API enhances the simulator's realism, allowing users to observe and react to dynamic cryptocurrency price fluctuations in INR. This feature is crucial for understanding the volatile nature of the crypto market and how trading decisions can be influenced by real-time data. The persistent storage of user data via MongoDB, unlike many transient simulators, adds another layer of practicality, enabling users to track their progress and review their simulated transaction histories over time.

While the simulator excels in demonstrating wallet operations and trading mechanics, its design intentionally excludes direct engagement with a live blockchain. This means aspects such as genuine decentralization, network consensus mechanisms, or the intricacies of gas fees are not part of the simulation. This limitation, however, is a deliberate trade-off that allows the platform to remain risk-free and focused on its primary educational objectives, avoiding the complexities and potential costs associated with blockchain interaction.

In conclusion, the Crypto Wallet Simulator successfully achieves its objective of providing a comprehensive and accessible educational tool for cryptocurrency literacy. By combining real-time API integration, robust cryptographic methods, and a sleek user interface, it effectively transforms abstract blockchain concepts into tangible, hands-on experiences. This secure and self-contained environment empowers learners to explore the world of digital currencies with confidence, ultimately fostering a deeper understanding of this rapidly evolving financial landscape.

7. REFERENCES

[1] I. Homoliak and M. Perešíni, “SoK: Cryptocurrency Wallets – A Security Review and Classification based on Authentication Factors,” *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–8, May 2024, doi: https://doi.org/10.1109/icbc59979.2024.10634439.

[2] Y. Erinle, Y. Kethepalli, Y. Feng, and J. Xu, “SoK: Design, Vulnerabilities, and Security Measures of Cryptocurrency Wallets,” *arXiv.org*, Aug. 04, 2023. https://arxiv.org/abs/2307.12874

[3] L. Zhou, Z. Liu, F. Zhang, and M. K. Reiter, “CrudiTEE: A Stick-and-Carrot Approach to Building Trustworthy Cryptocurrency Wallets with TEEs,” in *Leibniz Int. Proc. in Informatics (LIPIcs), * vol. 316, 6th Conf. on Advances in Financial Technologies (AFT), Sep. 2024, pp. 16:1–16:25, doi: 10.4230/LIPIcs.AFT.2024.16.

[4] A. Albshri, A. Alzubaidi, B. Awaji and E. Solaiman, "Blockchain Simulators: A Systematic Mapping Study," *2022 IEEE International Conference on Services Computing (SCC)*, Barcelona, Spain, 2022, pp. 284-294, doi: 10.1109/SCC55611.2022.00049.

[5] E. Rosa, G. D’Angelo, and S. Ferretti, “Agent-Based Simulation of Blockchains,” *Communications in Computer and Information Science*, pp. 115–126, 2019, doi: https://doi.org/10.1007/978-981-15-1078-6_10.

[6] Zheng, J.; Dike, C.; Pancari, S.; Wang, Y.; Giakos, G.C.; Elmannai, W.; Wei, B. An In-Depth Review on Blockchain Simulators for IoT Environments. *Future Internet* 2022, *14*, 182. https://doi.org/10.3390/fi14060182

[7] Zhu, J., & Zhang, L. (2023). “Educational Game on Cryptocurrency Investment: Using Microeconomic Decision-Making to Understand Macroeconomics Principles”. *Eastern Economic Journal*, *49*(2), 262–272. https://doi.org/10.1057/s41302-023-00240-7

[8] P. Medeiros and Leonidas Deligiannidis, “An Educational Guide to Creating Your Own Cryptocurrency,” *Transactions on computational science and computational intelligence*, pp. 163–177, Jan. 2021, doi: https://doi.org/10.1007/978-3-030-70873-3_12.