Decentralized Accounting System Using Blockchain Technology

Mrs. Suvarna Patil [1], Mayur Mohite [2], Harshal Gharat [3], Abhijeet Chavan [4], Anushka Kele [5]

[1 2 3 4 5] Marathwada Mitra Mandal's College Of Engineering, Pune, Maharashtra, India.
[1 2 3 4 5] Dept Of Computer Engineering, Pune, Maharashtra, India
[1] suvarnapatil@mmcoe.edu.in
[2] mayurmohite1562@gmail.com
[3] harshalgharat1918@gmail.com
[4] chavanabhijeet2003@gmail.com
[5] anushkakele1@gmail.com

Abstract - As digital transformation continues to reshape the accounting landscape, the limitations of traditional centralized systems such as vulnerability to fraud, lack of transparency, and inefficiencies in manual processes have become increasingly apparent. This research presents the real-world deployment of a blockchain-based accounting system on the Ethereum Sepolia testnet, emphasizing transparency, automation, and security in financial transactions. The system integrates essential accounting operations including inventory management, customer handling, sales processing, invoice generation, and payment execution, all powered by smart contracts.Users interact with the system through a web application, which connects to Ethereum-compatible wallets to initiate transactions. Each transaction is recorded on the blockchain through a structured flow triggering invoice creation, smart contract logic execution, and transaction validation across decentralized nodes. The use of the Sepolia testnet allowed for safe, cost-free testing of the smart contract logic, payment flows, and data immutability features without the risks associated with mainnet deployment. The deployment results show accurate transaction processing, successful invoice-to-payment traceability, and enhanced auditability, all without requiring third-party intermediaries. The system's architecture also demonstrates how decentralized applications (dApps) can automate accounting workflows in real-time, minimizing human errors and providing a transparent ledger accessible to all stakeholders. This practical validation confirms that blockchain technology is not only theoretically advantageous but also technically feasible and scalable for modern accounting environments.

***Key Words*:** Blockchain, Ethereum, Sepolia Testnet, Smart Contracts, Accounting System, Decentralized Applications, Transparency, Financial Automation, Data Integrity

1.INTRODUCTION

Accounting systems play a critical role in managing an organization's financial data, tracking sales, generating invoices, and ensuring regulatory compliance. Traditionally built on centralized architectures, these systems are often susceptible to data manipulation, lack transparency, and require extensive manual intervention. Such limitations can lead to inefficiencies, increased operational costs, and delayed reporting.

Blockchain technology offers a transformative alternative by introducing decentralization, immutability, and automation into financial systems [1][2]. By distributing data across a network of nodes, blockchain eliminates single points of failure and ensures secure, tamper-proof records. When paired with smart contracts self-executing programs that enforce predefined rules blockchain enables automated and verifiable transaction processing.

Building on our earlier conceptual work, this study presents the practical deployment of a blockchain-based accounting system on the Ethereum Sepolia testnet. Sepolia, a public test environment that mirrors Ethereum mainnet functionality, allows safe and cost-free validation of smart contracts and decentralized applications (dApps). Our system enables users to log in with crypto wallets, manage goods and customers, generate invoices, and process payments using smart contracts, with each transaction permanently recorded on the blockchain. This blockchain integration streamlines accounting by automating tasks, enhancing data integrity, and enabling real-time auditability. While challenges such as scalability and fluctuating gas fees remain, advancements like Ethereum 2.0 and Layer 2 solutions are steadily improving the ecosystem.

This paper explores the deployed system's architecture, smart contract functionality, and transaction outcomes on the Sepolia testnet. Through this implementation, we demonstrate blockchain's practical potential to reshape modern accounting systems into secure, efficient, and transparent platforms.

2. System Overview

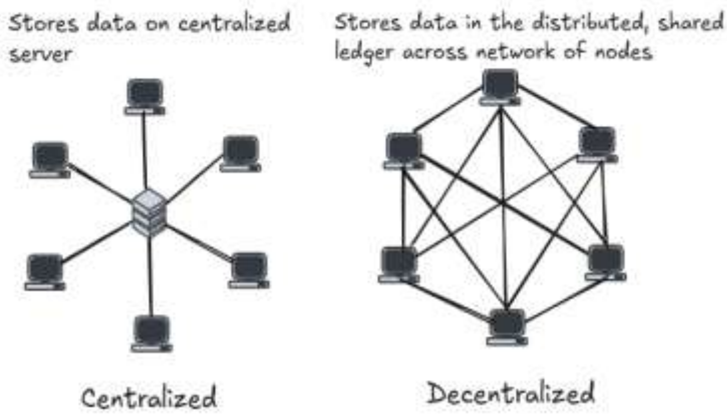


Fig -1: Centralized Vs Decentralized

The blockchain-based accounting system developed in this study addresses key financial management tasks, including inventory tracking, customer management, invoice generation, and secure payment processing. Traditional accounting systems, typically dependent on centralized databases, are vulnerable to unauthorized access and manipulation. In a centralized system, data is stored on a single server or controlled by a single entity, making it susceptible to hacking, data breaches, or even internal manipulation. Moreover, such systems often require trusted third parties to validate and authenticate transactions, which can introduce inefficiencies and additional costs.In contrast, this study introduces a decentralized alternative using the Ethereum blockchain and smart contracts to automate and secure accounting workflows. In a decentralized system, data is distributed across multiple nodes, making it nearly impossible to alter or tamper with the information without consensus from the network [4][5].

2.1 System Architecture

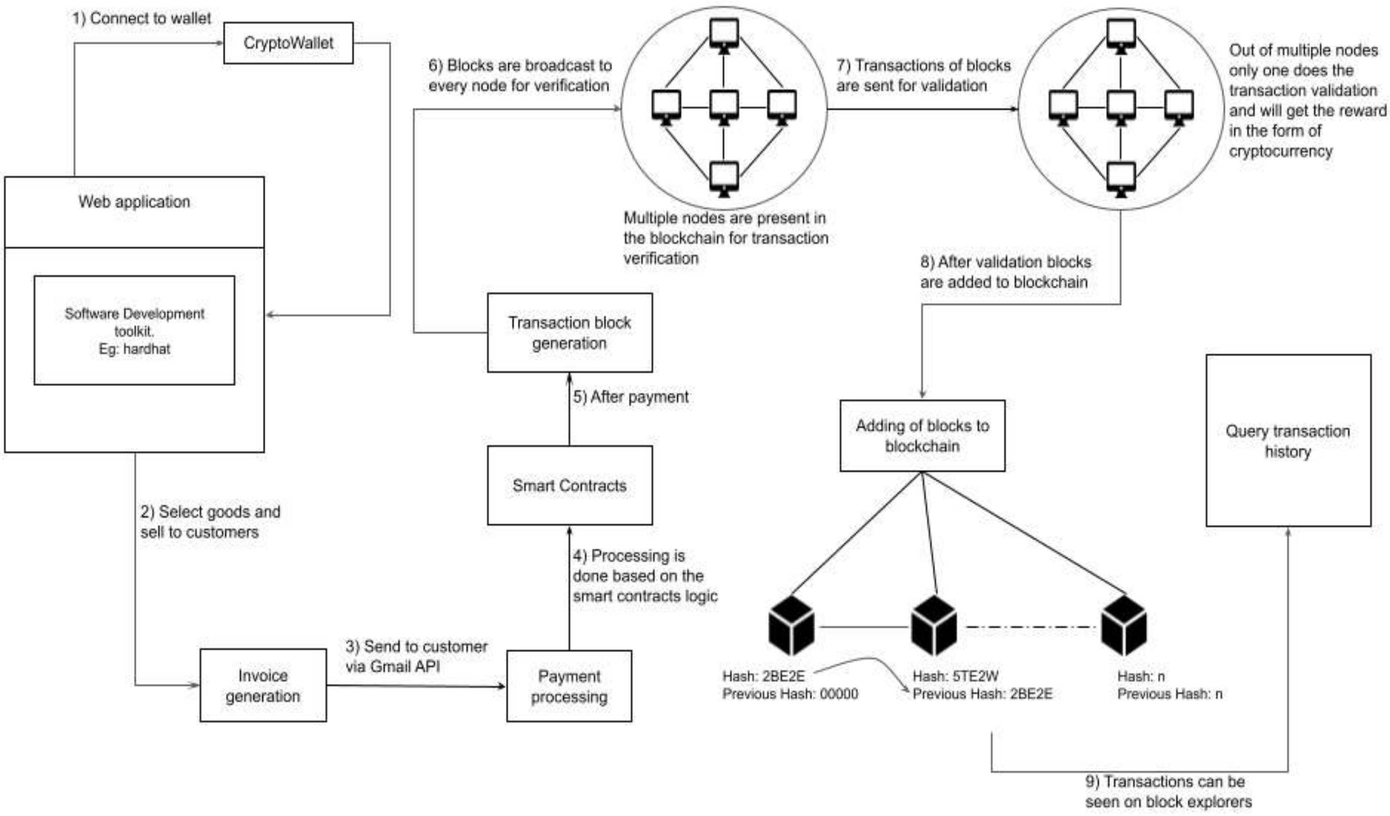


Fig -2: System Architecture

The architecture illustrated in the diagram represents a blockchain-based accounting system that integrates a web application with smart contracts and transaction validation on a decentralized blockchain network. The process begins when a user connects their crypto wallet to the web application. This wallet acts as the user's digital identity and is essential for authorizing transactions. Within the web application, which is developed using a software development toolkit such as Hardhat, the user selects goods to sell to customers. Once the goods are selected, an invoice is generated and automatically sent to the customer using the Gmail API.

This structure eliminates the need for third-party intermediaries, reduces the risk of fraud, and ensures transparency and accountability. The system integrates a user-friendly web interface with a decentralized backend for enhanced security and transparency. By leveraging Ethereum, all transactions are recorded on a public ledger, ensuring immutability and verifiability. Deployment on the Sepolia testnet allows for comprehensive validation of the platform's functionality using test Ether, providing a cost-effective way to test the system before moving to the mainnet.

This initiates the next step of the workflow, where the system processes payments through a secure payment processing module.Once payment is initiated, smart contracts come into play. These are pre-programmed agreements that execute specific instructions based on defined conditions.

The logic embedded within these smart contracts ensures that transactions are carried out correctly and transparently. After the smart contract processes the transaction, a new transaction block is generated. This block contains all the relevant payment and transaction details.

The newly generated block is then broadcast to every node in the blockchain network. Each node, essentially a computer participating in the blockchain, receives the block for verification. This ensures that the transaction is valid and has not been tampered with. Multiple nodes are involved in the verification process, but only one node is ultimately responsible for validating the transaction. This node, often selected through a consensus mechanism like proof of work or proof of stake, receives a reward in the form of cryptocurrency for its effort.

After a block is successfully validated, it is added to the blockchain. Each block contains a unique hash along with the hash of the previous block, ensuring a secure and immutable chain of records. This linked structure makes tampering virtually impossible, as altering one block would require changing every subsequent block. Once added to the blockchain, the transaction becomes part of the permanent ledger. At this stage, users can query the blockchain to view the transaction history, and these transactions are also publicly visible through blockchain explorers, offering transparency and accountability.

2.2 Deployment Environment

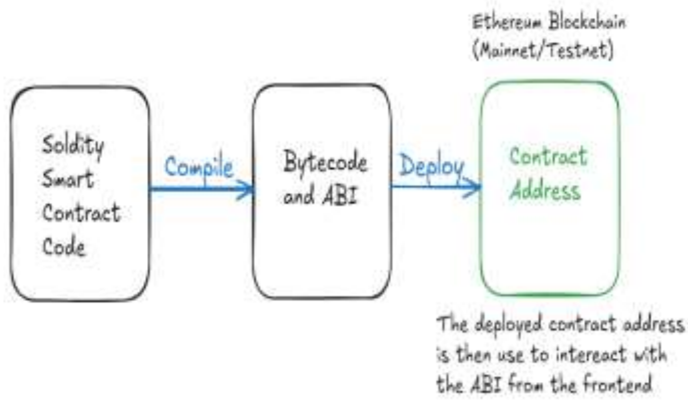


Fig -3: Contract Deployment

The blockchain-based accounting system was deployed on the Ethereum Sepolia testnet, a public network that supports the Proof-of-Stake (PoS) consensus mechanism. Sepolia is designed to replicate the behavior of the Ethereum mainnet while allowing developers to conduct robust testing without incurring real monetary costs. It uses test Ether to simulate transactions, making it an ideal environment for experimenting with and validating blockchain applications before deploying them to production. The use of Sepolia also enables testing under realistic network conditions, including block finality and transaction ordering, which is essential for a financial application like this.

The system's smart contracts were written in Solidity, Ethereum's core contract-oriented programming language. Solidity is well-suited for implementing custom business logic on the blockchain, and in this project, it was used to define how invoices are created, how payments are processed, and how transaction data is stored [6][7]. For managing the entire contract lifecycle, the team employed Hardhat, an advanced Ethereum development environment. Hardhat simplifies the process of compiling contracts, running automated tests, deploying to local and remote networks, and debugging smart contract behavior. Its plugin ecosystem also provides additional tools such as gas usage reporting and stack traces, which help identify inefficiencies and potential vulnerabilities during development.

To ensure secure user interaction with the blockchain, the system integrates MetaMask, a browser-based Ethereum wallet. MetaMask allows users to authenticate themselves by connecting their wallets to the application and sign transactions directly from their browser. This integration eliminates the need for users to manually enter private keys or sensitive information, thereby maintaining a strong security posture. All private keys are stored locally within MetaMask, ensuring that users retain full control over their credentials at all times.

The frontend of the system was developed using React.js, a widely adopted JavaScript library for building user interfaces. React enables the creation of a responsive and interactive web application that allows users to seamlessly manage inventory, register customers, generate invoices, and monitor transactions. Its component-based structure ensures code reusability and performance optimization, making it easier to maintain and scale the application in the future. The smooth integration between the React frontend, MetaMask, and the blockchain backend provides users with a real-time, intuitive experience that bridges traditional accounting functions with decentralized technology.

2.3 Workflow

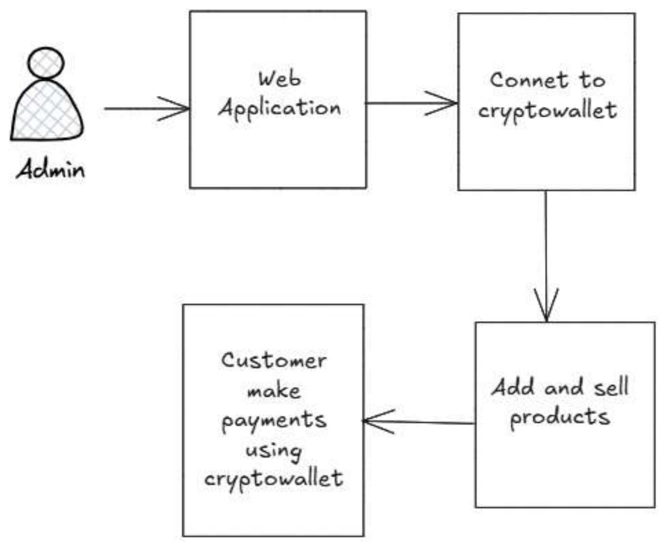


Fig -4: Workflow

The system is primarily operated by an admin, who authenticates securely via MetaMask, connecting their crypto wallet to the platform. This wallet-based login ensures that only authorized personnel can access and manage the system. Once logged in, the admin has full access to key operational features, including the ability to add and manage inventory items, register new customers, and initiate sales transactions.

When a sale is completed, the system automatically generates an invoice, capturing all relevant details such as the customer's information, purchased items, total amount, and payment status. This invoice is then securely recorded within the platform, ready for payment processing.

Customers make payments by interacting with MetaMask, which facilitates the transfer of test Ether to the business's address. Upon receiving the payment, the smart contract is triggered to validate the transaction ensuring the correct amount is received from the correct sender. Once validated, the smart contract updates the invoice status to "paid" and logs the transaction immutably on the blockchain.

The admin interface provides real-time updates of each transaction and maintains a complete history of all activities. This allows the admin to monitor payment confirmations, review past transactions, and maintain accurate financial records with ease. The overall workflow is designed to be efficient, secure, and transparent, combining traditional accounting operations with the benefits of decentralized technology.

2.4 Comparative Analysis with Traditional Systems

A comprehensive comparison between the blockchain-based accounting system and traditional accounting software reveals several critical advantages in terms of security, efficiency, transparency, and cost-effectiveness. Traditional systems typically operate on centralized databases managed by a single entity or organization [8]. This centralized architecture introduces multiple vulnerabilities, including the risk of unauthorized access, internal fraud, and single points of failure. Data in these systems can often be altered without detection, requiring significant manual auditing efforts to ensure accuracy and compliance.

Furthermore, conventional accounting processes usually depend on third-party intermediaries such as banks, auditors, or payment gateways to validate transactions, manage payments, and conduct audits. These intermediaries add complexity, increase operational costs, and introduce delays in financial workflows, particularly in invoice processing and payment settlement.

In contrast, the blockchain-based system developed in this study leverages decentralized ledger technology to enhance both transparency and security. All transactions are recorded immutably on the Ethereum blockchain, meaning once a transaction is confirmed, it cannot be altered or deleted. This feature alone drastically reduces the potential for fraud or error. The use of smart contracts allows for automated execution of accounting operations such as invoice generation, payment validation, and status updates without the need for human intervention. This not only streamlines processes but also ensures that transactions occur precisely as programmed, reducing delays and manual oversight.

Additionally, the system provides real-time tracking of all financial activities. As soon as a transaction is initiated and confirmed on the blockchain, it is reflected immediately in the user interface, giving the admin access to up-to-date information without waiting for batch processing or manual reconciliation. This immediate visibility improves decision-making and operational responsiveness.

The transparency inherent in public blockchains also enhances trust among stakeholders. All transactions can be independently verified via blockchain explorers, eliminating disputes over transaction authenticity or status. For organizations, this provides a built-in audit trail that significantly reduces the burden of compliance and financial reporting.

2.5 Challenges Encountered

While the blockchain-based accounting system demonstrated strong potential and functionality, the development journey presented several challenges that required careful navigation and problem-solving. One of the most persistent issues was transaction latency on the Sepolia testnet, particularly during times of network congestion. Although the Sepolia network is highly useful for testing, it doesn't always provide consistent performance under load. This led to delays in payment confirmations, which, while tolerable in a test environment, would be unacceptable in a real-world business scenario where time-sensitive transactions are critical.

Another significant challenge was related to user onboarding and accessibility. The system relies on MetaMask for wallet integration and test Ether for executing transactions. While these tools are standard in the blockchain ecosystem, new users especially those without a technical background often found the setup process daunting. Tasks such as installing the MetaMask extension, creating a wallet, and obtaining test Ether from faucets proved to be obstacles for smooth adoption. These friction points suggest that for wider adoption, future iterations of the platform will need more streamlined onboarding processes and user-friendly guidance.

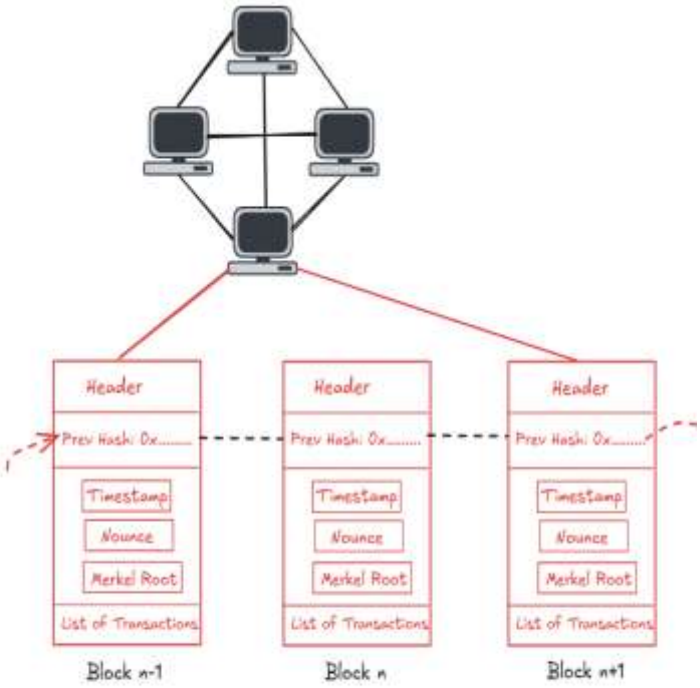
From a technical standpoint, Ethereum's gas constraints and transaction size limits posed significant design considerations. The smart contracts had to perform complex accounting operations, including invoice creation, validation, and status tracking. Balancing this logic within the constraints of gas limits required code optimization, modular design, and rigorous testing. Inefficient code could lead to failed transactions or excessive costs, even on a testnet.

Security remained a top priority throughout the project. Given the financial nature of the platform, any vulnerability could be catastrophic. We conducted multiple rounds of testing and code review to identify and patch potential weaknesses such as reentrancy attacks, integer overflows, and unauthorized access points. We implemented common Solidity security best practices and used tools for static analysis and simulated attacks. This process was time-intensive and required a deep understanding of both the Ethereum Virtual Machine (EVM) and Solidity programming.

Additionally, working with off-chain and on-chain data coordination introduced another layer of complexity. Since sensitive customer and invoice details were stored off-chain for privacy, we had to ensure accurate linking between the blockchain ledger and the external data storage using cryptographic hashes and event tracking. Any inconsistency here could lead to mismatched records, undermining trust in the system.

These experiences underscored the importance of building not just secure and functional smart contracts but also developing scalable infrastructure, intuitive UX, and robust support tools. Moving forward, addressing these issues through Layer 2 integrations, gas-efficient smart contract patterns, and improved user documentation will be essential to making blockchain-based accounting systems viable for real-world enterprise deployment.

2.6 Security and Privacy Considerations



The security and privacy of the system are fundamentally supported by Ethereum's robust cryptographic architecture. Every transaction on the platform is digitally signed using the private key of the sender's wallet, ensuring that only authorized individuals can initiate and approve actions. These signatures provide authenticity and non-repudiation, making unauthorized tampering virtually impossible.

Smart contracts are also written with secure coding practices in mind to protect against known vulnerabilities such as reentrancy attacks, integer overflows, and unauthorized access. Extensive testing and validation are performed during development to ensure that contracts behave exactly as intended.

To balance transparency with data privacy, the system uses a hybrid storage model [10]. Sensitive business and customer information such as itemized invoice details or customer addresses is not stored directly on the blockchain. Instead, only cryptographic references (hashes) of this data are recorded on-chain. This approach maintains the integrity and verifiability of the data while ensuring that confidential information is kept off the public ledger.

A key technique used to achieve this is hashing. Hash functions convert data into fixed-length strings that are unique to the original content. Even a small change in the original data results in a completely different hash, making it easy to detect tampering. These hashes are stored on the blockchain as proofs that the corresponding data existed at a certain time and in a certain state.

For larger datasets or grouped transactions, the system can also utilize a Merkle tree structure. A Merkle tree is a hierarchical data structure where each leaf node is a hash of a data block, and each parent node is the hash of its child nodes. The top-most hash, called the Merkle root, summarizes all the underlying data in a single hash value. This Merkle root can be stored on-chain, allowing the system to prove the integrity of a large dataset without storing all of it on the blockchain.

2.7 Future Enhancements

Looking ahead, several key enhancements are planned to improve the scalability, usability, and functionality of the blockchain-based accounting system. A major focus is the integration of Ethereum Layer 2 solutions such as Optimism or zk-Rollups, which would significantly reduce gas fees and increase transaction throughput [12]. These technologies allow transactions to be processed off-chain and then batched onto the Ethereum mainnet, making the system more cost-effective and efficient, particularly for businesses with high transaction volumes. Additionally, the user experience will be enhanced by offering mobile-friendly wallet options, enabling users to interact with the platform seamlessly from their mobile devices. The introduction of QR code-based payment systems will further streamline the payment process, allowing customers to easily make payments by scanning a QR code with their mobile wallets, thus reducing manual errors and improving convenience. To address the volatility risks associated with Ether, the platform may also introduce stablecoin support such as USDC or DAI offering businesses a more stable and predictable currency for transactions. Finally, features like role-based access control and advanced reporting tools will be introduced to enhance the system's functionality for larger businesses and organizations, allowing for more granular control and better analytics for financial management. These upgrades will ensure that the system can scale to meet the needs of growing businesses while maintaining a user-friendly experience and robust security.

3. CONCLUSION

In conclusion, the blockchain-based accounting system developed in this study offers a revolutionary alternative to traditional accounting software. By leveraging the transparency, immutability, and security of the Ethereum blockchain, the system provides a more efficient, reliable, and secure way to handle financial transactions. It eliminates the risks associated with centralized databases, reducing the potential for fraud and unauthorized access. With its user-friendly interface and seamless integration with MetaMask for secure authentication, the system makes it easy for businesses to manage inventory, sales, and payments while keeping all data safely recorded on the blockchain.

While the system has shown great potential during testing on the Sepolia testnet, there are still challenges to overcome, such as improving scalability, reducing transaction costs, and enhancing user experience. Looking forward, the integration of Layer 2 solutions, mobile-friendly features, stablecoin support, and more advanced functionalities will help make the system even more practical for businesses of all sizes.

Ultimately, this system has the potential to redefine financial management, offering businesses a more efficient, transparent, and secure way to operate. With further refinement and optimization, it could pave the way for blockchain adoption in the accounting world, transforming how financial transactions are conducted and recorded on a global scale.

REFERENCES

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
2. Zheng Z,)Ge. S.. Daii Cik•n. & Wary, H. (2018). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. IEEE International Congress on Big Data.
3. Wood, G. (2014). Ethereum: A Secure Decentralized Generalized Transa&n Ledger. Etfæreum Project Yellow Paper.
4. Buterin, V. (2014). A Next-Generation Smart Contract Decentralized Applicatk)n Platform.
5. Christidis, K, & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE
6. Dannen, C (2017). Introducing Ethereum and Solidity. A press.
7. Morkunas, V. J.J Paschen, & Boon, E. (2019). How Blockchain Technologies Impact Your Business Model. Business Horizons.
8. Tapscott, & Tapscott. A (2016). Blockchain Revolution. Penguin.
9. A. M. (2017). Mastering Bitcoin: Programming the Open Blockchain. O'Reilly Media.
IO. Crosby. M., Pattanayak, Verma, S.. & Kalyanaraman, V. (2016). Blockchain Technology: Beyond Bitcoin. Applied Innovation Review, 2.
11. Zhang. xue, & Liu, L. (2019). Security and Privacy on Blockchain.ACM Computing Surveys (CSUR).
12. Gluchowski, B. (2021). Introduction to zk-RolhJps. Ethereum Foundation Blog.
13. De Angelis, S.. Aniello, L, Baldoni, Lombardi, F., & Margheri, A. (2018). PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain. Italian Conference on Cybersecurity.
14. Kuo, T.-T% H.-E, & Ohno-Machado, L (2017). Blockchain Distributed Ledger Technologies for Biomedical and Health Care Applications. Journal of the American Medical Informatics Association.
15. Gatteschi V.v Lamberti, F., Demartini, C.. Pranteda, C, & Santamaria. V. (2018). Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough? Future Internet, 10(2).20.

BIOGRAPHIES

Hi, I'm Mayur Mohite, a passionate Software Development Engineer with a strong background in building scalable and responsive web applications using JavaScript, TypeScript, React, and Next.js. I have hands-on experience in developing smart contracts with Solidity, integrating blockchain functionality into modern web solutions. I thrive in Agile team environments, where I focus on writing clean, efficient code with an emphasis on performance, usability, and security. I'm always eager to learn new technologies and contribute to impactful, forward-thinking projects.