**CHATHUB USING BLOCKCHAIN**

*A synopsis submitted for partial*

*Fulfillment for award of the degree of*

**Master of Computer Application**

By

**Saikat Das**

**Chayan Shaw**

**Sayan Ghosh**

**Chirasree Chaki**

**Gargi Das**

**Under the guidance of**

**Mr. Sudip Das**

**Narula Institute of Technology**



*May,2025*

**Narula Institute of Technology**

**81, Nilgunj Road, Agarpara**

**Kol-700109**

**West Bengal**

**CIRTIFICATE OF ORIGINALITY**

We have developed this project as fulfillment of master in computer application 4th semester (Sub Code: MCA 596) while developing the project no unfair means or illegal copies of software etc. have been used neither any part of this project nor any documentation have been submitted elsewhere or copied.

|  |  |
| --- | --- |
| **Name** | **Roll no.** |
| **Saikat Das** | **434123120056** |
| **Chayan Shaw** | **434123120057** |
| **Sayan Ghosh** | **434123120050** |
| **Chirasree Chaki** | **434123020051** |
| **Gargi Das** | **434123120055** |

**CERTIFICATE OF APPROVAL**

This is to certify that the project entitled **“ChatHub”** has been carried out by MCA 4th semester students.

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll No.** | **Registration No.** |
| **SAIKAT DAS** | **434123010056** | **231270510035** |
| **CHYAN SHAW** | **434123010057** | **231270510011** |
| **SAYAN GHOSH** | **434123010050** | **231270510040** |
| **CHIRASREE CHAKI** | **434123020051** | **231270510012** |
| **GARGI DAS** | **434123010055** | **231270510017** |

It is understood that by this approval the undersigned do not necessarily endorse any of the statements made or opinion expressed in their project. But we approve it only for the purpose for which it is submitted.

**Dept. Of CA**

**&**

**Project Guide**

Narula Institute of Technology

Agarpara, Kolkata - 700109

**ACKNOWLEDGEMENT**

This project would not have been successful without the sincere motivation, cooperation and guidance of our mentor **AST. PROF. SUDIP DAS**, Dept. of Computer Application, NARULA INSTITUTE OF TEHNOLOGY who has provided us with useful resources and motivation through the various phases of this project which has proven to be crucial for the completion of this documentation.

We have a special thanks to **PROF.SUBHASREE BHATTACHARJEE**, and other faculties of CA department. The guidance and support received from all the members who contributed and who are contributing to this project, is vital for the success of the project. We are grateful for their constant support and help

Saikat Das

Chayan Shaw

Sayan Ghosh

Chirasree Chaki

Gargi Das

**ABSTRACT**

Decentralized application make use of peer-to-peer networks, this ensures that no network failure can occur due to central node failure. Blockchain serves as an immutable ledger which allows messaging to take place in a decentralized manner. A decentralized application for communication and resource sharing is need in today’s world, where keeping data on a centralized server can be risky and costly experience. With the help of various consensus, we can implement different ways to share resources and communicate. Together with Blockchain and Decentralized Applications, we can create a secure and reliable messaging application that overcomes the drawbacks of traditional messaging applications.

**INDEX**

* **Introduction**
* **Objectives**
* **Motivation**
* **Novelty**
* **Related Work**
* **Proposed Approaches**
* **System Requirement Specification**
* **Feasibility Study**
* **System Requirements**
* **Data Flow Diagram**
* **ER Diagram**
* **Use Case Diagram**
* **Project Scheduling**
* **Gantt & PERT Charts**
* **Working Procedure**
* **Result & Discussion**
* **Sample Code**
* **UI Design**
* **Conclusion**
* **Future Scope**
* **Bibliography**

**INTRODUCTION**

In today’s generation chatting over messaging platforms are a part of an individual’s lifestyle. Today’s most of the communication happens over social media platforms. All these platforms also provide users the option to share multimedia attachments leveraging their communication protocols over sockets. All these chat or messaging platforms are processed through centralized servers. All the user’s message or information (maybe confidential) is being processed by the central server before transmitting the same to intended recipients. The issue with these kinds of system is that all the information are visible at processing servers even if the messages or information transmitted are claimed to be end to end encrypted. The author has created a messaging or rather say a simple chat application and has explained experimentally shown how the transmitted messages are visible at processing servers. Nevertheless, the system of the centralized system has scalability issues when compared to decentralized computing systems. In this work, the author has proposed a blockchain based solution based on ethereum platform using Whisper Protocol to the issues that exist in traditional messaging or chat applications.

**OBJECTIVES**

Blockchain is an emerging technology platform for developing decentralized applications and data storage, over and beyond its role as the technology underlying the cryptocurrencies. The basic tenet of this platform is that it allows to create a distributed and replicated ledger of events, transactions, and data generated through various IT processes with strong cryptographic guarantees of tamper resistance, immutability, and verifiability. The global blockchain market size is expected to grow from USD 3.0 billion in 2020 to USD 39.7 billion by 2025. The objective of this course is to provide conceptual understanding of how blockchain technology can be used to innovate and improve business processes. The course covers the technological underpinning of blockchain operations in both theoretical and practical implementation of solutions using blockchain technology.

**MOTIVATION**

A chat app built on blockchain technology offers several compelling motivations, primarily centered around enhanced security, privacy, and user control. By leveraging blockchain's decentralized nature, these apps aim to create a more secure and transparent communication environment, free from censorship and third-party control.

Here's a deeper look at the key motivations:

***1.Enhanced Security and Privacy:***

Blockchain's immutability and encryption ensure that messages are secure and cannot be tampered with after they are recorded. End-to-end encryption further protects user data, allowing only the intended recipient to access the message.

***2.Decentralization and User Control***:

Unlike traditional chat apps that rely on centralized servers, blockchain-based apps distribute control among users, making them more resilient to censorship and single points of failure. This empowers users to have greater autonomy over their data and interactions.

***3.Transparency and Traceability:***

Blockchain's transparent and traceable nature allows users to verify the integrity of their communication history, fostering trust and accountability.

**NOVELTY**

While there are encrypted chat apps like Signal and Telegram, most still rely on centralized servers and traditional user authentication (e.g., phone numbers, emails). Blockchain introduces self-sovereign identity and decentralized message routing, which can offer:

• True anonymity and ownership of identity using decentralized identifiers (DIDs).

• Token-based access control, eliminating centralized moderation.

• Immutable message receipts, useful for audit trails in legal, enterprise, or diplomatic contexts.

• End-to-end encryption combined with on-chain identity verification, enabling trustless communication.

Potential Features:

1. Wallet-based Login (e.g., MetaMask, Phantom).

2. End-to-end Encrypted Messaging with metadata stripped using zero-knowledge proofs.

3. On-chain Reputation System tied to NFTs or Soulbound Tokens.

4. Decentralized Storage (e.g., IPFS, Arweave) for message history.

5. Token Rewards for content moderation or message validation via DAO governance.

Applications:

• Anonymous whistleblower networks.

• DAO communication tools.

• DeFi customer support chat systems.

• Censorship-resistant journalism communication platforms.

**RELATED WORK**

***1. Decentralized Chat Application Using Blockchain***

• Explore how to build a peer-to-peer chat system with no central server using blockchain (e.g., IPFS + Ethereum).

• Focus on message immutability, privacy, and censorship resistance.

***2. End-to-End Encrypted Messaging with Blockchain-Based Identity Verification***

• Use blockchain for verifying user identities or public keys while actual messaging is encrypted and off-chain.

• Solve key exchange trust issues.

***3. Token-Based Incentives in Chat Applications***

• Introduce crypto-token rewards for active users, helpful responses, or moderation efforts.

• Explore a DAO-based governance model.

**PROPOSED APPROACHES**

***1. Hybrid On-Chain/Off-Chain Messaging System***

Goal: Balance scalability, cost-efficiency, and security.

• Messages stored off-chain (e.g., IPFS or a decentralized storage layer).

• Blockchain stores hashes of messages, timestamps, sender/receiver IDs for verification.

• Smart contracts manage message receipts, proofs, or disputes.

***2 Blockchain-Based Identity and Authentication***

Goal: Secure, decentralized user authentication and public key sharing.

• Use DIDs (Decentralized Identifiers) or ENS for usernames.

• Identity data and public keys registered on blockchain.

• Eliminates need for centralized login/auth servers.

***3. Incentive-Driven Chat with Token Economy***

Goal: Encourage active participation, moderation, or content curation.

• Integrate a token reward system (ERC-20 or similar).

• Earn tokens through contributions (helpful messages, moderation).

• Tokens used for tipping, premium features, or governance voting.

***4. Full Decentralized Chat via Smart Contracts***

Goal: Maximize decentralization (no servers, censorship-resistant).

• Use Ethereum or Layer 2 chains to store short messages.

• Message contracts handle sending, storing, and reading logic.

• High cost and low speed—good for experiments, not production use.

***5. Privacy-Preserving Messaging with zk-SNARKs***

Goal: Protect metadata and message content even from blockchain observers.

• Implement zk-SNARKs or zk-STARKs to prove message delivery without revealing content.

• Combine with mixers or stealth addresses to obscure sender/receiver.

***6. DAO-Governed Messaging App***

Goal: Community-controlled chat environment.

• Users vote on moderation rules, bans, or features via a DAO.

• Token-holders have voting power.

• Can use Snapshot + smart contracts for proposal execution.

***Software Requirement Specification***

**FEASIBILITY STUDY**

1. ***Technical Feasibility***

*Setup Process:* Using Ganache and MetaMask ensures a user-friendly development and testing environment. The ability to deploy smart contracts with Truffle and integrate the frontend with a local blockchain network adds significant flexibility.

*Scalability Concerns:* While the current setup (using Ganache) is sufficient for development, scalability issues may arise in production. Transitioning to a Layer 2 solution (e.g., Polygon) could address transaction speed and gas fee challenges.

*Decentralization:* The use of IPFS for message storage ensures decentralized content storage, enhancing reliability.

***2. Economic Feasibility***

*Cost Efficiency:* Ganache provides a free, local blockchain environment for development. However, deploying to Ethereum's mainnet may incur high gas fees, especially for frequent transactions. Consider using a cost-effective testnet or sidechain like Polygon or Binance Smart Chain.

*Potential Monetization:* Token-based rewards or governance structures (integrated via ERC-20 tokens) can provide monetization options while fostering user engagement.

1. ***Social Feasibility***

*Privacy and Data Ownership:* The app empowers users with control over their data, addressing growing concerns about privacy breaches on centralized platforms.Its decentralized nature eliminates reliance on third-party servers, ensuring confidentiality.

*Freedom of Expression:*The app is resistant to censorship, providing a platform for unrestricted communication. This is especially valuable for individuals in regions with strict communication regulations.

*Community-Driven Development:*

By leveraging DAO-based governance, users can actively participate in decision-making processes, fostering a sense of ownership and trust in the platform.

*Accessibility for Marginalized Communities:*

The app’s secure and censorship-resistant design makes it an ideal tool for marginalized groups, journalists, and activists, enabling safe and confidential communication.

*Adoption Challenges:*

While blockchain has immense potential, mainstream adoption may face hurdles due to lack of awareness or reluctance from non-tech-savvy individuals. User education initiatives could bridge this gap.

*Trust Building:*

Blockchain’s transparent and immutable features enhance trust among users, as it ensures that communication and data cannot be tampered with.

*Social Inclusion:*

With global reach and decentralized architecture, the app can promote inclusive communication, eliminating geographical barriers and providing a secure space for diverse communities.

**SYSTEM REQUIREMENTS**

***Hardware Requirements***

Processor: Minimum Intel Core i5 or AMD Ryzen 5 (Recommended: i7/Ryzen 7 or higher

RAM: At least 4GB (Recommended: 8GB or more for better performance)

Storage: SSD (32GB minimum) for faster read/write speeds (Recommended: 64 GB SSD or higher)

GPU: Not mandatory, but a dedicated GPU (e.g., NVIDIA GTX 1650 or higher) can help with cryptographic computations

Internet: Stable high-speed internet for blockchain synchronization

***Software Requirements***

Operating System: Windows 10/11, macOS, or Linux (Ubuntu recommended)

Blockchain Frameworks: Ethereum (Solidity), Hyperledger Fabric, or Polkadot

Smart Contract Development: Solidity, Rust, or Go

***Development Tools:***

Node.js (for backend services)

Web3.js or Ethers.js (for blockchain interactions)

IPFS/Filecoin (for decentralized storage)

Hardhat/Truffle (for smart contract deployment)

Database: NoSQL (MongoDB) or decentralized storage (IPFS, Arweave)

Security Tools: AES-256 encryption, zk-SNARKs for privacy

***Additional Dependencies***

Wallet Integration: MetaMask, WalletConnect

Consensus Mechanism: Proof-of-Stake (PoS) or Proof-of-Authority (PoA)

Testing Frameworks: Mocha, Chai, Jest (for smart contract testing)

**DATA FLOW DIAGRAM**

***Check User Login***

***Chat Management***

***Generate Chat History***

***Login Management***

**CHAT HUB**

***Generate Chat History Report***

***Chat History Management***

***Generate Chat Profile***

***Chat Profile Management***

*DFD Level 1*

**LOGIN**

**Manage Modules**

USER

**Forgot Password**

**Check Credentials**

***Manage Chat Details***

***Manage Chat Profile Details***

**Change Password**

***Manage Chat History***

***Manage Sender/Receiver***

*DFD Level 2*

**ER DIAGRAM**

**Recipient Name**

**MailID**

**Sender Name**

**User**

**Message**

**Sends**

**Msg ID**

**Status**

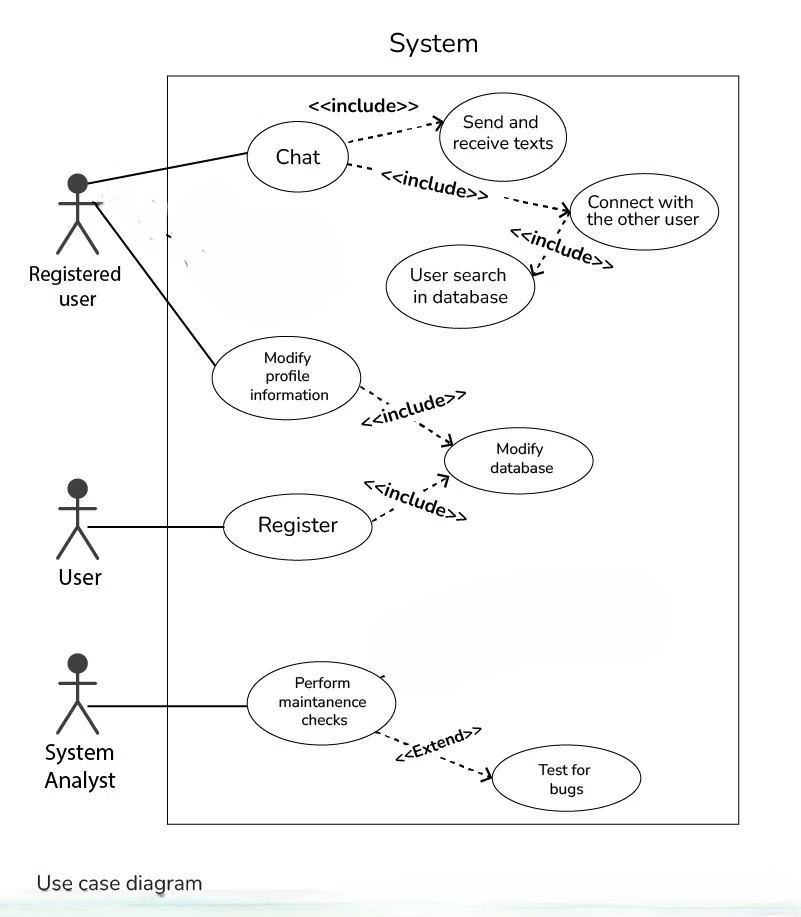
**Username**

**Time Stamp**

**Password**

*ER Diagram of CHAT HUB*

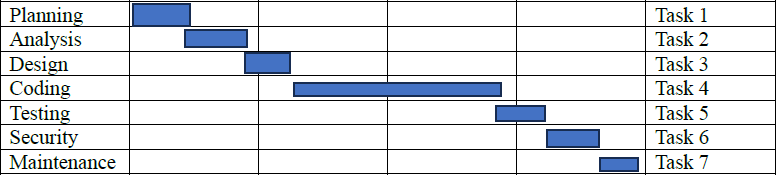
**USE CASE DIAGRAM**

****

*Use Case Diagram*

***Project Scheduling***

**GNATT CHART**

****

**PERT CHART**

|  |  |  |  |
| --- | --- | --- | --- |
| TASKS | ACTIVITIES | DAYS | DEPENDENCIES |
| 1 | Planning | 7 | - |
| 2 | Analysis | 5 | Task 1 |
| 3 | Design | 10 | Task 2 |
| 4 | Coding | 25 | Task 3 |
| 5 | Testing | 7 | Task 4 |
| 6 | Security | 8 | Task 5 |
| 7 | Maintenance | 4 | Task 4 |

**WORKING PROCEDURE**

***1. Define Project Scope***

• Decide the goal: decentralized messaging, privacy, user identity, token system, etc.

• Identify features:

• One-to-one Messaging

• User authentication (Web3 login)

• Message verification (hashing or storing on-chain)

• Token rewards, moderation, etc.

***2. Choose Tech Stack***

Frontend:

• React / React Native (for web & mobile)

• Wallet integration: MetaMask, WalletConnect

Backend:

• IPFS / Filecoin / Ceramic (for message or file storage)

• Node.js or Python (if any off-chain services are needed)

Blockchain Layer:

• Ethereum (or Layer 2: Polygon, Optimism)

• Smart contracts (Solidity)

• Optional: Lens Protocol, Lit Protocol, XMTP for messaging

***3. Design System Architecture***

• Decide what goes on-chain vs. off-chain:

• Public keys, message hashes = on-chain

• Message content = IPFS/off-chain

• Use smart contracts for:

• Message validation

• Token rewards / governance

• Reputation or moderation system

***4. Develop Smart Contracts***

• Write in Solidity (or Vyper).

• Examples:

• sendMessage(sender, receiver, messageHash)

• registerUser(wallet, username)

• rewardUser(userAddress, tokens)

• Test using Hardhat or Truffle.

• Deploy on testnet (Mumbai, Goerli, etc.)

***5. Frontend Integration***

• Connect wallet via Web3 libraries.

• Let users:

• Register/login with wallet

• Send messages

• Read from blockchain/IPFS

• Display verified message history

***6. Decentralized Storage Integration***

• Use IPFS to store message content (optional encryption).

• Store IPFS hash in smart contract.

***7. Testing***

• Test:

• Smart contract logic (unit + integration)

• Wallet interactions

• Performance on different networks

***8. Deployment***

• Deploy smart contracts to mainnet or a Layer 2 chain.

• Host frontend (e.g., Vercel, IPFS, or Fleek).

• Optionally open source the repo.

***9. Future Upgrades (Optional)***

• Add token economy (reward active users).

• DAO-based governance.

• Zero-knowledge privacy layer.

• Voice or video integration

**RESULT AND DISCUSSION:**

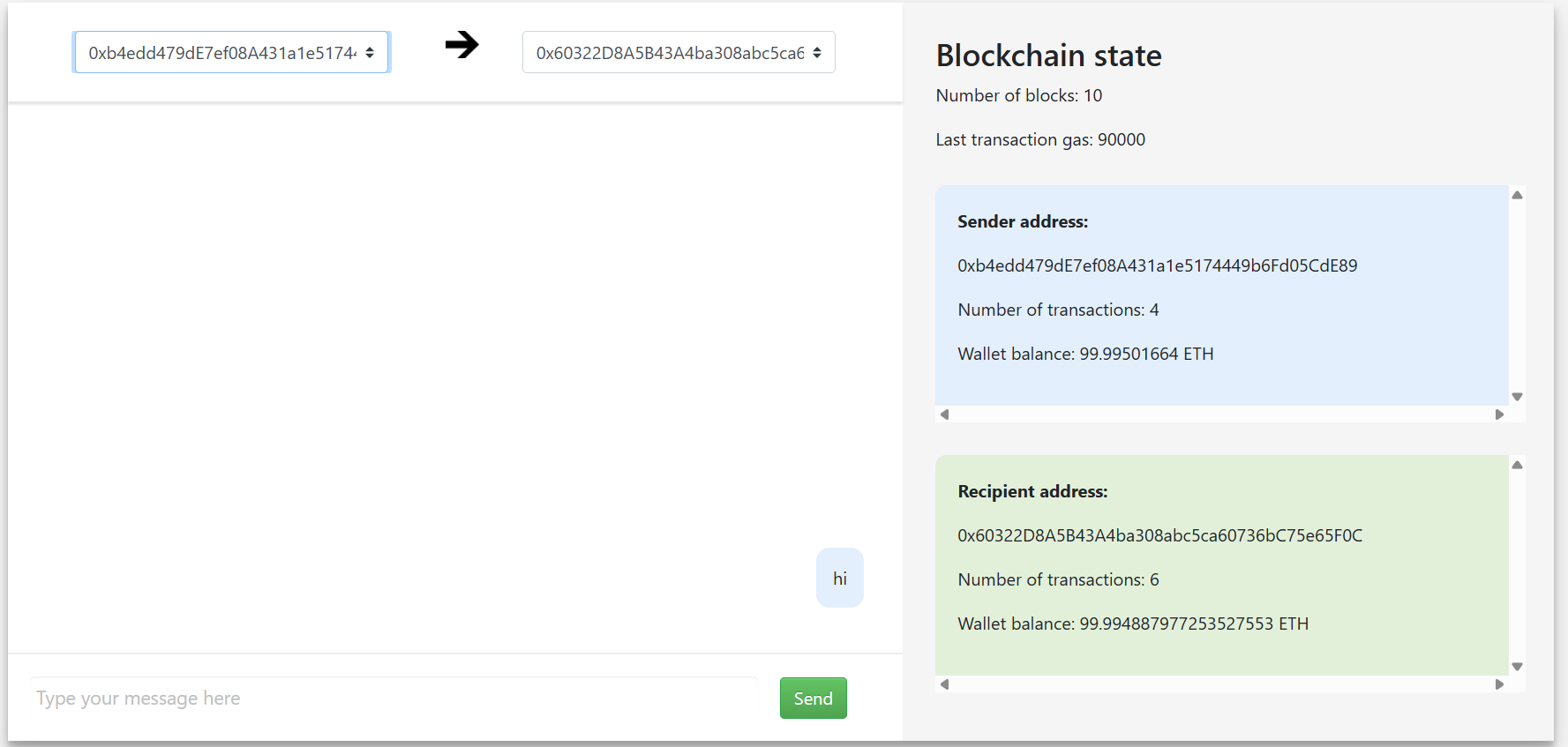
The blockchain-based chat app was successfully developed using Ethereum smart contracts and IPFS for decentralized messaging. Users could log in with a crypto wallet, register a username, and send messages. Message content was stored on IPFS, while hashes and metadata were stored on-chain for integrity.

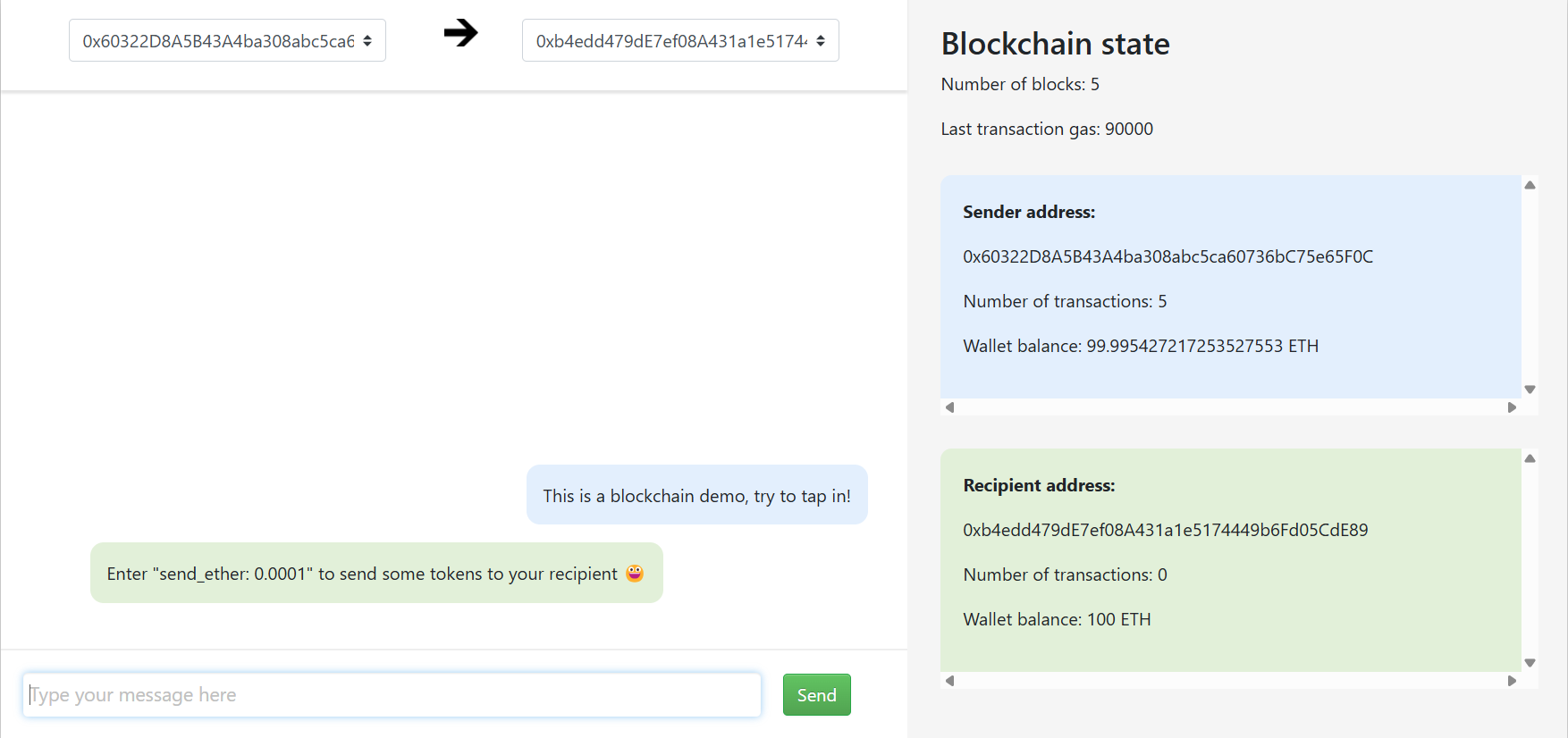
The system ensured immutability and censorship resistance, but had limitations in scalability, latency, and cost, especially when sending messages due to blockchain transaction times. Off-chain storage helped reduce gas fees, though real-time performance was limited. Privacy of message content was maintained, but sender and receiver addresses remained public.

Future improvements include moving to Layer 2 for lower fees, adding end-to-end encryption, and exploring DAO-based moderation.

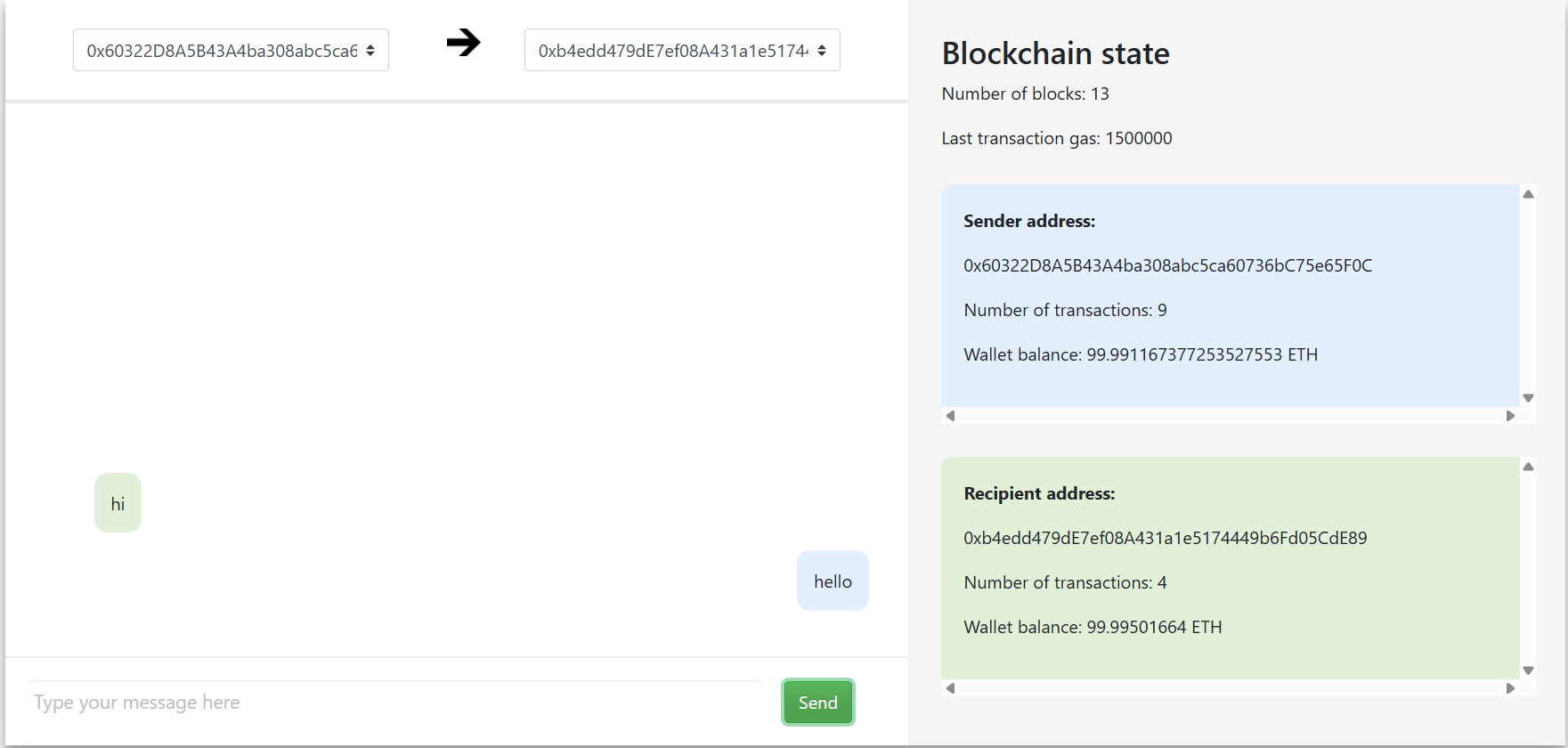
**SAMPLE CODE**

**UI DESIGN**

*****Chat Hub*



*Sender*

****

*Recevier*

**CONCLUSION**

Blockchain is a powerful tool for resolving complex issues quickly. Its ability to provide security in an open environment makes it attractive for usage in a variety of other fields, including health care, IoT applications, and finance. E-commerce retailers and delivery partners can use consortium blockchains to avoid fraud during transit by continuously updating package positions on the blockchain. One of the most innovative potential uses of blockchain could be to avoid fraud in chit funds, which are used to save money in Indian society. It can also serve as a ledger for disadvantaged farmers to share resources. We give a state-of-the-art survey of blockchain technology in this study. We began by discussing the background, classification, architecture, and several sorts of consensus.

Blockchain technology strengthens messaging apps by introducing decentralization and transparency. Each message or metadata entry is securely logged on the blockchain, creating an immutable record that is resistant to tampering.

**FUTURE SCOPE**

**1. Enhanced Privacy:**

**-** Zero-knowledge proofs (zk-SNARKs) for metadata protection.

- Stealth addresses for sender/receiver anonymity.

**2. Improved Scalability:**

**-** Use Layer 2 solutions (e.g., Polygon) for faster transactions.

- Adopt sharding or multi-chain architecture**.**

**3. Multimedia Support:**

**-** Encrypted sharing of images, audio, and video via IPFS or Arweave.

**4. Token Economy:**

**-** Reward users for active participation and governance voting.

**5. DAO Governance:**

- Community-driven decision-making for rules and feature updates.

**BIBLIOGRAPHY**

***Websites:***

Ethereum Foundation- https://ethereum.org/

Ganache Project- https://archive.trufflesuite.com/ganache/

ConsenSys- "How Blockchain Can Power Decentralized Messaging." -https://consensys.net/blog/blockchain-explained/how-blockchain-can-power-decentralized-messaging/

MetaMask- https://metamask.io/ — (for blockchain-based app integrations)

***Tools/Frameworks:***

1.Solidity Documentation-https://docs.soliditylang.org/

2.Web3.js Documentation-https://web3js.readthedocs.io/