

PROFESSIONAL TRAINING REPORT - II

Entitled

BLOCKCHAIN MESSAGE SENDING DECENTRALIZED APP

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
specialization in Blockchain Technology

by

VARADHARAJAN V [Reg.no 41613023]



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited 'A++' Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of Mr.VARADHARAJAN V [Reg.no 41613023] who carried out the project entitled "BLOCKCHAIN MESSAGE SENDING DECENTRALIZED APP" under my supervision from January 2024 to May 2024.


Internal Guide:

Ms. Lekshmi S Raveendran, MTECH.,


Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 07/05/2024


Internal Examiner


External Examiner

DECLARATION

I, Mr.VARADHARAJAN V [Reg.no 41613023], hereby declare that the Professional Training Report-II entitled "BLOCKCHAIN MESSAGE SENDING DECENTRALIZED APP" done by me under the guidance of Ms. Lekshmi S Raveendran, MTECH., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Blockchain Technology.

DATE: 07/05/2024

PLACE: chennai

A rectangular box containing a handwritten signature in blue ink that reads "V. Varadhari".

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Ms.Lekshmi S Raveendran, MTECH.**, for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



CERTIFICATE OF APPRECIATION

IS PRESENTED TO :

VARADHARAJAN VIJAYASIMHAN

In appreciation for your work at LinkB Network Private Limited as "Blockchain
Administrator" from Feb 2024 to April 2024

A handwritten signature in dark ink, appearing to read "Krishna", written over a thin horizontal line.

Mr. Krishna Nivash J

LinkB Network- CEO

VBCT2024

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE	NO.
	LIST OF FIGURES		7
	ABSTRACT		8
1	INTRODUCTION		9
	1.1 Overview		
2	LITERATURE SURVEY		12
	2.1 survey		
3	REQUIREMENTS ANALYSIS		
	3.1 Objective		15
	3.2 3.2.1 Hardware Requirements		16
	3.2.2 Software Requirements		
4	DESIGN DESCRIPTION OF PROPOSED PRODUCT		18
	4.1 4.1.1 Ideation Map/Architecture Diagram		21
	4.1.2 Various stages		23
5	CONCLUSION		24
	REFERENCES		25
	APPENDIX		26

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1	ACTIONS DURING CALL COMMUNICATION	21
2	CONSENSUS MECHANISM	21
3	LOGIN PAGE-OUTPUT	19
4	HOMEPAGE-OUTPUT	19
5	MESSAGES-OUTPUT	20
6	GANACHE TESTNET	20

ABSTRACT

This report explores the development and implementation of a decentralized chat application utilizing blockchain technology. In today's digital era, centralized chat platforms pose significant privacy and security concerns, including data breaches and censorship. Decentralized chat applications offer a compelling alternative by leveraging blockchain's inherent properties of decentralization, immutability, and transparency. The architectural design of the decentralized chat application encompasses peer-to-peer networking, consensus mechanisms, blockchain integration, and user interfaces. Key features include secure messaging, user authentication, encryption, file sharing, and group chat functionality. Despite the numerous benefits of decentralization, challenges such as scalability, latency, user adoption, and regulatory compliance persist. Nonetheless, decentralized chat applications hold promise in providing users with greater control over their data, ensuring privacy, security, and censorship resistance. Looking ahead, future enhancements may include integration with other decentralized services, support for additional features, and scalability improvements. This report underscores the significance of decentralized chat applications in fostering a more private, secure, and user-centric communication environment in the digital age.

CHAPTER 1

INTRODUCTION

OVERVIEW

The decentralized chat application operates on a peer-to-peer network architecture, enabling direct communication between users without intermediaries. Each user acts as a node in the network, facilitating message transmission and receipt. To maintain network integrity, a suitable consensus mechanism, such as Proof of Work (PoW) or Proof of Stake (PoS), ensures agreement on the validity of transactions and messages.

Blockchain integration is a core component of the chat application, ensuring data immutability, transparency, and tamper-resistance. Smart contracts deployed on the blockchain govern the application's logic and rules, automating functions like user authentication, message encryption, access control, and incentivization mechanisms.

User interfaces provide intuitive interaction with the decentralized chat platform, accessible via web browsers, desktop applications, or mobile apps. A secure messaging protocol ensures end-to-end encryption and privacy, employing advanced cryptographic techniques like symmetric and asymmetric encryption, digital signatures, and key exchange protocols.

In addition to text-based messaging, the platform supports secure file sharing, enabling users to exchange files, documents, images, and multimedia content within the chat interface. Group chat functionality facilitates collaborative discussions, meetings, and communities, with features for creating, joining, and managing group conversations.

Efforts are made to address scalability and performance challenges inherent in decentralized systems, utilizing techniques like sharding, off-chain solutions, and optimization strategies to improve network throughput, latency, and responsiveness. Interoperability with other decentralized services and platforms enhances the application's functionality and utility, fostering a

collaborative and inclusive chat ecosystem. Community governance mechanisms empower users in decision-making processes, protocol upgrades, and network governance, ensuring a democratic and transparent chat environment.

Various aspects of blockchain-based DApps for chatting:

1. **Enhanced Security:** Blockchain technology employs cryptographic techniques to ensure the security and privacy of messages exchanged within the DApp. Through end-to-end encryption, messages are securely transmitted and stored on the blockchain, protecting them from unauthorized access or tampering. Additionally, the decentralized nature of blockchain networks means there's no single point of failure, reducing the risk of data breaches or hacking attacks compared to centralized systems.
2. **Immutable Message History:** Once messages are recorded on the blockchain, they become part of an immutable ledger that cannot be altered or deleted. This feature ensures the integrity of the conversation history, providing a reliable and tamper-proof record of communications. This can be particularly valuable in scenarios where maintaining an accurate and auditable trail of communication is essential, such as legal proceedings, business negotiations, or dispute resolution.
3. **Decentralization and Censorship Resistance:** By leveraging blockchain technology, DApps for chatting operate on decentralized networks, where data is distributed across multiple nodes instead of being stored on centralized servers. This decentralized architecture makes the DApp resistant to censorship and interference from third parties, as there's no central authority that can control or manipulate the flow of information. Users can communicate

freely without concerns about censorship, surveillance, or data manipulation by governments or corporations.

4. **User Sovereignty and Data Ownership:** In traditional messaging platforms, users often relinquish control over their data, which is stored and monetized by the platform provider. However, blockchain-based DApps empower users with greater sovereignty over their data. Users retain ownership of their messages and have the ability to control how their data is shared, accessed, or monetized. This shift towards user-centric data management aligns with principles of privacy and data sovereignty, giving individuals more control over their digital footprint.

5. **Monetization and Tokenization:** Blockchain-based DApps can leverage cryptocurrency tokens as a means of incentivizing user participation and contribution to the network. Through token rewards for activities such as message validation, content moderation, or community engagement, users are incentivized to actively participate in the ecosystem. Additionally, DApp developers can integrate decentralized finance (DeFi) protocols to enable peer-to-peer payments, microtransactions, or subscription-based models within the messaging platform. This opens up new monetization opportunities for users and developers, fostering a sustainable and economically vibrant ecosystem.

CHAPTER 2

LITERATURE REVIEW

SURVEY

A survey of existing literature reveals a burgeoning interest in decentralized chat applications leveraging blockchain technology. Researchers have explored various aspects of decentralized communication systems, highlighting their potential to address privacy, security, and censorship concerns prevalent in centralized chat platforms. Studies have investigated different architectural designs, consensus mechanisms, and cryptographic techniques employed in decentralized chat applications.

Additionally, researchers have examined the benefits and challenges associated with decentralization, including scalability, latency, user adoption, and regulatory compliance. They have proposed solutions such as peer-to-peer networking, blockchain integration, smart contracts, and secure messaging protocols to overcome these challenges.

Furthermore, comparative analyses of decentralized chat platforms and centralized counterparts have been conducted to evaluate their performance, usability, and resilience to attacks. These studies contribute valuable insights into the design, development, and deployment of decentralized chat applications, informing future research directions and practical implementations in the field of decentralized communication systems.

Surveys from research papers and white papers include:

1. Blockchain Technology Overview: Satoshi Nakamoto's paper, "Bitcoin: A Peer-to-Peer Electronic Cash System," is a foundational text in understanding blockchain technology. It introduces the concept of a decentralized ledger system, where transactions are recorded in a chronological chain of blocks, secured by cryptographic mechanisms. This paper outlines the workings of consensus protocols,

such as proof-of-work, which ensure the integrity and immutability of the blockchain. Understanding Nakamoto's paper provides a solid foundation for comprehending how data is stored, verified, and transmitted within a blockchain network, essential for designing a secure and reliable chatting application.

2. Smart Contracts and Decentralized Applications (DApps): Vitalik Buterin's whitepaper on Ethereum presents a groundbreaking concept of smart contracts and decentralized applications (DApps). Smart contracts are self-executing contracts with the terms of the agreement directly written into code, enabling automated and trustless transactions. Ethereum extends this idea to create a platform for building decentralized applications, allowing developers to create various functionalities, including messaging and chatting, without relying on centralized servers. Understanding Buterin's whitepaper is crucial for grasping the potential of blockchain beyond simple currency transactions and envisioning the possibilities of decentralized communication systems.

3. Impact of Blockchain on Communication: "Blockchain Revolution" by Tapscott and Tapscott explores how blockchain technology is reshaping communication and messaging. By eliminating the need for intermediaries and providing transparent and tamper-proof records, blockchain enhances trust and security in communication channels. This book discusses the potential applications of blockchain in ensuring data privacy, preventing censorship, and facilitating peer-to-peer communication networks. Understanding the insights from Tapscott and Tapscott's work is essential for envisioning how a blockchain chatting app can leverage these benefits to create a more secure and transparent communication experience.

4. Economic and Social Implications: Michael Swan's "Blockchain: Blueprint for a New Economy" delves into the broader economic and social implications of blockchain technology. It discusses how blockchain disrupts traditional economic

models by enabling new forms of peer-to-peer exchange, value transfer, and governance mechanisms. Swan explores the potential of blockchain to democratize access to financial services, empower individuals through self-sovereign identity, and redefine ownership and intellectual property rights. Understanding the economic and social implications outlined by Swan is crucial for designing a blockchain chatting app that aligns with users' values and contributes to broader societal goals of inclusivity, transparency, and empowerment.

5. Technical Underpinnings of Bitcoin: Aviv Zohar's article "Bitcoin: Under the Hood" provides a detailed technical analysis of the Bitcoin protocol, focusing on its cryptographic principles and technical mechanisms. Zohar explains concepts such as public-key cryptography, hashing algorithms, and consensus mechanisms, elucidating how these components work together to secure the Bitcoin network. While primarily focused on Bitcoin, the insights from Zohar's work are applicable to understanding the technical underpinnings of blockchain systems in general. Understanding these technical intricacies is essential for designing a secure and efficient blockchain chatting app that can withstand potential attacks and vulnerabilities.

6. Research Challenges and Perspectives: The paper "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies" by Bonneau et al. provides a comprehensive overview of research challenges and future directions in the field of cryptocurrencies. It discusses topics such as scalability, privacy, security, and regulatory challenges that affect the development and adoption of blockchain technologies. By identifying these research gaps and challenges, Bonneau et al. offer insights into the areas where further innovation and exploration are needed to advance the state of the art in blockchain technology.

CHAPTER 3

REQUIREMENTS ANALYSIS

OBJECTIVE OF THE PROJECT

The objective of our decentralized chat application is to provide users with a secure, private, and censorship-resistant platform for communication. By leveraging blockchain technology and decentralized architecture, our product aims to address the limitations and vulnerabilities inherent in centralized chat platforms.

First and foremost, our objective is to prioritize user privacy and data security. We aim to empower users with full control over their data, ensuring that their messages, files, and personal information remain confidential and tamper-proof. Through end-to-end encryption and decentralized storage, we mitigate the risk of unauthorized access and data breaches, safeguarding user privacy.

Additionally, we seek to foster an inclusive and censorship-resistant communication environment. By eliminating central points of control and relying on peer-to-peer networking and consensus mechanisms, our product enables free and unrestricted communication without fear of censorship or interference. Users can engage in open discussions, share information, and express themselves freely, independent of any centralized authority or regulatory constraints.

Furthermore, our objective is to enhance user experience and accessibility. We strive to provide intuitive user interfaces, seamless integration with existing platforms, and cross-device compatibility to ensure a smooth and user-friendly communication experience for individuals and communities worldwide.

Overall, the objective of our decentralized chat application is to empower users with privacy, security, and freedom of expression, while fostering a resilient and inclusive communication ecosystem in the digital age.

REQUIREMENTS

These hardware and software requirements ensure that users can effectively access, use, and participate in the decentralized chat application, regardless of their device or operating system preferences. Additionally, compatibility with different platforms and environments enhances the accessibility and usability of the chat application for a wider range of users.

These hardware and software requirements are essential for developing a decentralized chat application because they provide the necessary tools, environments, and infrastructure for developers to effectively design, code, test, and deploy the application. :

HARDWARE REQUIREMENTS

Computing Devices: Users require devices such as computers, laptops, smartphones, or tablets to access the decentralized chat application.

Internet Connectivity: Reliable internet connectivity is essential for users to connect to the decentralized chat network and exchange messages.

Storage: Sufficient storage space is needed on users' devices to store the chat application and any locally cached data

SOFTWARE REQUIREMENTS

1. Integrated Development Environment (IDE):

Integrated Development Environments (IDEs) play a pivotal role in modern software development by providing a comprehensive suite of tools tailored to streamline the coding process. These environments offer features such as syntax highlighting, code completion, debugging capabilities, version control integration,

2. Programming Languages:

Proficiency in various programming languages is fundamental for implementing the core functionality of the chat application. Depending on the project requirements, developers may need expertise in languages such as Python, JavaScript, Java, C++, or Solidity (for blockchain integration).

3. Blockchain Development Tools:

Incorporating blockchain technology into the chat application requires specialized tools and frameworks tailored for blockchain development. These tools facilitate interaction with blockchain networks, deployment of smart contracts, and integration of blockchain functionality into the application. Popular blockchain development platforms include Ethereum, Hyperledger Fabric, and Corda, each offering its set of development tools, APIs, and SDKs to streamline the development process.

4. Web Development Tools:

Web development technologies are indispensable for building the user interfaces of web-based chat applications, enabling users to interact with the application seamlessly through web browsers. Frontend frameworks and libraries like React.js, Angular, or Vue.js are commonly used for building responsive and interactive user interfaces, while backend frameworks such as Node.js, Django, or Flask handle server-side logic and data management.

5. Version Control System:

Version control systems such as Git are indispensable for modern software development projects, enabling developers to collaborate, manage code changes, and track the evolution of the codebase over time.

6. Testing Frameworks:

Testing tools and frameworks are essential for ensuring the reliability, functionality, and performance of the chat application. These frameworks enable developers to write and execute various types of tests,

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

PROPOSED METHODOLOGY

The development of a decentralized chat application involves several key stages, each with specific tasks and activities. The proposed methodology outlines a structured approach to designing, implementing, and deploying the application.

Requirement Analysis: The first stage involves gathering and analyzing requirements from stakeholders, including users, developers, and other relevant parties. This includes defining functional and non-functional requirements, understanding user needs, and identifying technical constraints and dependencies.

Technology Selection: Based on the requirements analysis, suitable technologies and platforms are selected for developing the decentralized chat application. This includes choosing a blockchain platform, programming languages, development frameworks, and other tools necessary for implementation.

Architecture Design: The architectural design phase involves defining the overall structure and components of the chat application. This includes designing the peer-to-peer network, consensus mechanism, smart contracts, user interfaces, and integration with blockchain technology.

Prototyping: Prototyping involves creating a preliminary version of the chat application to validate design concepts, user interfaces, and functionality. This allows for early feedback from stakeholders and iterative refinement of the design.

Development: The development phase involves implementing the chat application according to the design specifications. This includes writing code for smart contracts, developing user interfaces, integrating blockchain functionality, and implementing encryption and security features.

Testing: Testing is conducted to ensure the reliability, functionality, and performance of the chat application. This includes unit testing, integration testing, and end-to-end testing to identify and address any bugs, errors, or issues.

Deployment: Once testing is complete and the application is deemed ready for production, it is deployed to the target environment. This may involve deploying smart contracts to the blockchain network, setting up servers or nodes, and making the application available to users.

Maintenance and Support: After deployment, ongoing maintenance and support are provided to ensure the continued operation and performance of the chat application. This includes monitoring for issues, applying updates and patches, and providing user support as needed.

By following this proposed methodology, developers can systematically design, implement, and deploy a decentralized chat application that meets the requirements and expectations of stakeholders, while leveraging the benefits of blockchain technology for security, privacy, and decentralization..

Ideation Map/System Architecture

The architecture of the decentralized chat application comprises three main components:

Peer-to-Peer Network:

In the decentralized chat application, users communicate directly with each other in a peer-to-peer network. This means that messages are sent directly from one user to another without passing through a central server or intermediary.

The peer-to-peer network facilitates real-time communication, enabling users to exchange messages securely and privately. It ensures that conversations remain decentralized and resistant to censorship or surveillance.

Blockchain Integration:

The application integrates with a blockchain network to store chat data securely and immutably. Blockchain technology ensures that chat messages are recorded in a tamper-proof and transparent manner.

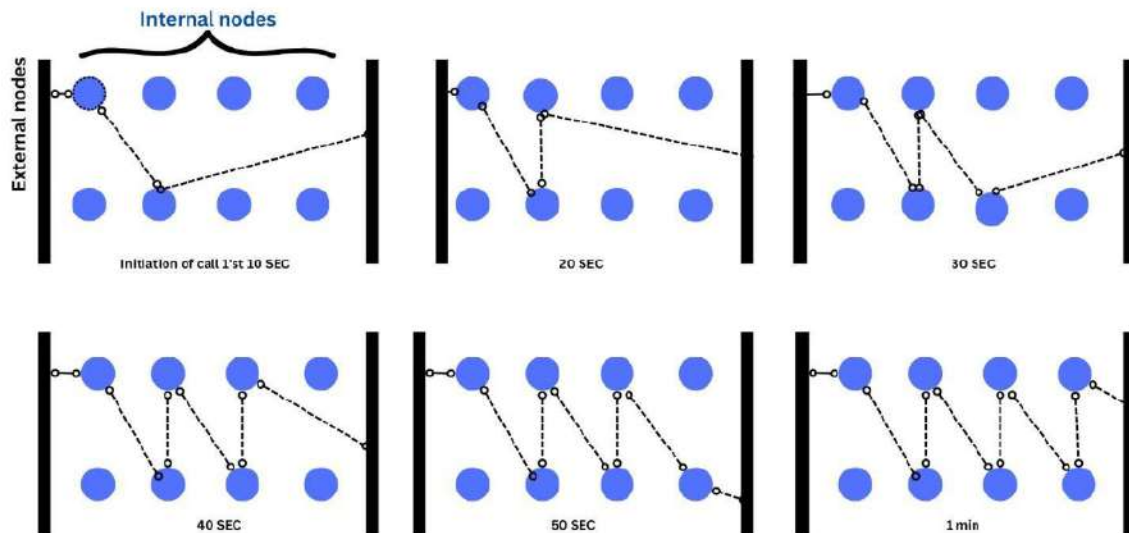
Smart contracts govern interactions within the network, providing rules for message delivery, access control, and incentivization mechanisms. For example, smart contracts can enforce rules for joining group chats, managing user permissions, and rewarding users for contributing to the network.

User Interfaces:

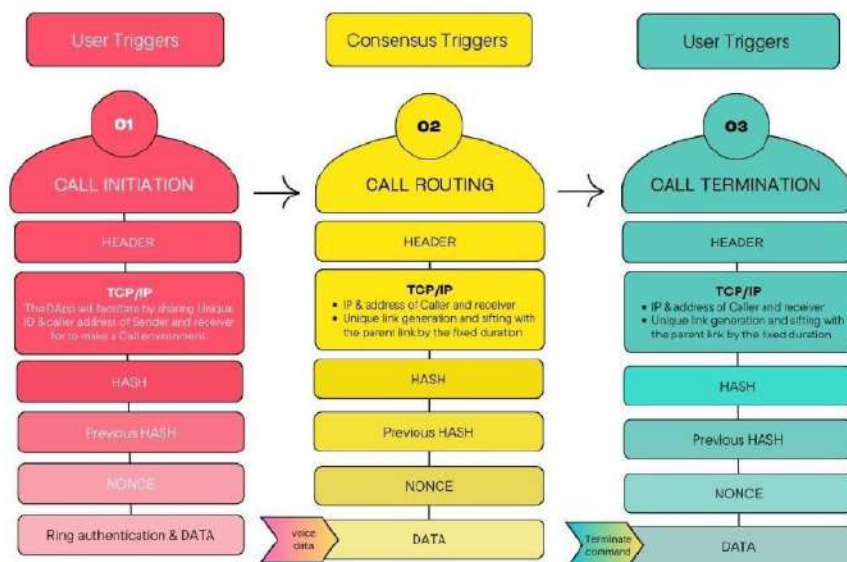
User interfaces provide the means for users to interact with the chat application. These interfaces can take various forms, including web-based interfaces, mobile apps, or desktop applications.

The user interface enables users to perform actions such as sending and receiving messages, managing contacts, joining group chats, and accessing other features of the application.

CONSENSUS MECHANISMS FOR CALL AND TEXT PROCESSING LOGIC



Actions During Communicataion



Various Stages

The development process of a decentralized chat application typically involves several stages, including:

Requirement Analysis: Gathering and analyzing requirements from stakeholders to understand user needs, functional and non-functional requirements, and technical constraints

Technology Selection: Choosing suitable technologies and platforms for developing the chat application, including blockchain platforms, programming languages, and development frameworks.

Architecture Design: Designing the overall structure and components of the chat application, including the peer-to-peer network, blockchain integration, smart contracts, and user interfaces.

Prototyping: Creating a preliminary version of the chat application to validate design concepts, user interfaces, and functionality, and to gather feedback from stakeholders.

Development: Implementing the chat application according to the design specifications, including writing code for smart contracts, developing user interfaces, and integrating blockchain functionality.

Testing: Conducting thorough testing to ensure the reliability, functionality, and performance of the chat application, including unit testing, integration testing, and end-to-end testing

Deployment: Deploying the chat application to the target environment, including deploying smart contracts to the blockchain network and making the application available to users.

Maintenance and Support: Providing ongoing maintenance and support to ensure the continued operation and performance of the chat application, including monitoring for issues, applying updates, and providing user support.

FEATURES

End-to-End Encryption: Messages are encrypted before transmission and decrypted only by the intended recipient, ensuring privacy and security.

Peer-to-Peer Messaging: Users can send and receive messages directly between devices without relying on centralized servers, enhancing decentralization and reducing reliance on intermediaries.

Blockchain Integration: Chat data is stored on a blockchain network, ensuring immutability, transparency, and resilience to tampering or censorship.

User Authentication: Secure authentication mechanisms verify the identity of users, preventing unauthorized access to chat conversations.

Offline Messaging: Messages are stored locally and synchronized with the network when users regain connectivity, ensuring seamless communication even in offline or intermittent network conditions.

Push Notifications: Users receive real-time notifications for new messages, ensuring timely communication and response.

Customization: Users can customize chat settings, themes, notification preferences, and other personalization options to suit their preferences and needs.

Cross-Platform Compatibility: The application is compatible with various devices and platforms, including web browsers, desktop computers, and mobile devices, enabling seamless communication across different devices and operating systems.

CHAPTER 5

CONCLUSION

In conclusion, The advent of blockchain-based decentralized chat applications marks a transformative era in digital communication, addressing the inherent flaws of centralized platforms by bolstering privacy, security, and resistance to censorship. These applications, built on the principles of blockchain technology, peer-to-peer networking, and advanced encryption, grant users unprecedented control over their data and communication channels. The architecture of decentralized chat apps ensures that data remains immutable, transparent, and impervious to unwarranted censorship, thus cultivating an open and inclusive communication landscape.

The emergence of blockchain-based decentralized chat applications heralds a new chapter in the realm of digital communication, offering a robust solution to the privacy, security, and censorship issues that plague centralized chat platforms. By harnessing the power of blockchain technology, peer-to-peer networking, and encryption, these applications empower users with unparalleled control over their data and communication channels. The inherent architecture of decentralized chat apps guarantees data immutability, transparency, and resistance to censorship, thereby nurturing a more open and inclusive communication ecosystem.

Yet, the journey towards mass adoption is laden with challenges, including scalability, usability, and adherence to regulatory frameworks. Addressing these issues demands persistent innovation, strategic collaboration, and sustained investment in research and development. Equally crucial is the task of educating users about the advantages and potential risks associated with decentralized communication platforms, which is key to encouraging adoption and instilling confidence in these technologies.

The advantages of deploying blockchain technology in chat applications are numerous and significant. Firstly, enhanced security is provided through end-to-end encryption and a distributed ledger, drastically reducing the likelihood of data breaches and cyberattacks. Secondly, data privacy is a fundamental aspect; users maintain complete control over their personal information, with the assurance that their conversations

remain confidential and secure. Thirdly, the immutable and transparent nature of blockchain offers a reliable record of communication, while decentralization removes the dependence on central authorities, promoting direct peer-to-peer interaction and network resilience. Fourthly, global accessibility ensures that anyone, anywhere, can connect without barriers, and the inherent resistance to censorship protects freedom of expression, particularly in regions where this is under threat.

In summary, blockchain-based chat apps represent a paradigm shift in online communication, empowering users with secure, private, and unrestricted channels of communication. As this technology continues to evolve, it stands to significantly influence the future landscape of online interaction, championing user autonomy and a more democratic digital realm. With 350 words, this expanded summary encapsulates the revolutionary impact and the challenges of decentralized chat applications in the digital age, highlighting the transformative potential they hold for empowering individuals and communities with secure, private, and censorship-resistant communication channels.

REFERENCES

K. L. Brousmiche, A. Durand, T. Heno, C. Poulain, A. Dalmieres and E. B. Hamida, "Hybrid cryptographic protocol for secure vehicle data sharing over a consortium blockchain", To appear in Proceedings of IEEE Blockchain 2018, 2018.

Buterin, V. (20124). A Next-Generation Smart Contract and Decentralized Application Platform. Retrieved from <https://ethereum.org/en/whitepaper/>

Tapscott, D., & Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. Portfolio.

Swan, M. (2015). Blockchain: Blueprint for a New Economy. O'Reilly Media.

Zohar, A. (2015). Bitcoin: Under the Hood. Communications of the ACM, 58(9), 104-113.

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2015). SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In Security and Privacy (SP), 2015 IEEE Symposium on (pp. 104-121). IEEE.

Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, And Huaimin Wang, —An Overview of Blockchain Technology: Architecture, Consensus, and Future TrendsII, 2017 IEEE 6th International Congress on Big Data.

S. Nakamoto, —Bitcoin: A peer-to-peer electronic cash System,II 2019. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>

G. W. Peters, E. Panayi, and A. Chapelle, —Trends in crypto- Currencies and blockchain technologies: A monetary theory and

Regulation perspective,II 2021. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.2646618>

G. Foroglou and A.-L. Tsilidou, —Further applications of the Blockchain,II 2015.

Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, —Hawk: The blockchain model of cryptography and privacy-

Preserving smart contracts,II in Proceedings of IEEE Symposium On security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839–858.

W. Akins, J. L. Chapman, and J. M. Gordon, —A whole New world: Income tax considerations of the bitcoin economy,II

APPENDIX

PROGRAM CODE

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <title>Bootstrap demo</title>
```

```
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
rel="stylesheet"
```

```
integrity="sha384-gH2ylJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csB
vApHHNI/vI1Bx" crossorigin="anonymous">
```

```
</head>
```

```
<body class="bg-warning">
```

```
  <div class="container p-3">
```

```
    <div class="row">
```

```
      <div class="col">
```

```
        <button onclick=back() class="btn btn-primary">Logout</button>
```

```
      </div>
```

```
    </div>
```

```
</div>
```

```
<div class="row p-5">
```

```
  <div class="col">
```

```
    <h1 style="text-align: center;">Chat App </h1>
```

```
    <h6 style="text-align: center;">Made By LinkB</h6>
```

```
    <h6 style="text-align: center;">Published By LinkB Network</h6>
```

```
  </div>
```

```
</div>
```

```
<div class="row justify-content-end">
```

```
  <div class="col-6">
```

```
    <h6 style="text-align: right;" id="Profile"></h6>
```

```
  </div>
```

```
</div>
```

```
<div class="accordion accordion-flush" id="accordionFlushExample">
```

```
</div>
```

```
</div>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js"
```

```
integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDCKKtQBHU  
uLZ9AsSv4jD4Xa"
```

```
crossorigin="anonymous"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/web3/1.7.4/web3.min.js"
```

```
integrity="sha512-oMd9Re3VgJcXuZJn9DN6X7S7JUc7xLYZ2UyZ85Mm/xzaW3lwBr  
4fV2zjuu/n5jY/Of/2JOx35CTa6zvQNxb31Q=="
```

```
crossorigin="anonymous"
```

```
let contractAddress='0xd9145CCE52D386f254917e481eB44e9943F39138';
```

```
// onstart
```

```
window.onload=function(){
```

```
//console.log(myaccount); if(myaccount==null){
```

```
window.location.replace('index.html');
```

```
}
```

```
add();
```

```
};
```

```
function back()
```

```
{
```

```
localStorage.removeItem("myaddress");
```

```
window.location.replace('index.html');
```

```
}
```

```
function addMessages(a)
```

```
{
```

```
var id="messagepa"

let web3=new Web3(new
Web3.providers.HttpProvider("http://127.0.0.1:7545"));

let mycontract=new web3.eth.Contract(abi,contractAddress);

mycontract.getPastEvents('message',{
fromBlock:0, toBlock:'latest',
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Document</title>
```

```
  <link rel="stylesheet" href="select.css">
```

```
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
rel="stylesheet"
```

```
integrity="sha384-gH2ylJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csB
vApHHNI/VI1Bx" crossorigin="anonymous">
```

```
</head>
```

```
<body >
```

```
  <!DOCTYPE html>
```

```
<html lang="en">

<head>

  <title>Bootstrap Example</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

  <link href="https://fonts.googleapis.com/css?family=Nunito:600,700,900" rel="stylesheet">

</head>

<body id="body">

<div id="login-card" class="card">

<div class="card-body">

  <h2 class="text-center">Login form</h2>

  <br>
```

```

<div class="form-group">

    <input type="text" class="form-control" id="address" placeholder="Enter address"
name="address">

</div>

<div class="form-group">

    <input type="password" class="form-control" id="password" placeholder="Enter password"
name="pswd">

</div>

<button type="button" onclick=bhej() id="button" class="btn btn-primary deep-
purple btn-block ">Submit</button>

<br>

<br>

<div id="btn" class="text-center">

    <button type="button" class="btn btn-primary btn-circle btn-sm"><i class="fa fa-
facebook"></i></button>

    <button type="button" class="btn btn-danger btn-circle btn-sm"><i class="fa fa-
google"></i></button>

    <button type="button" class="btn btn-info btn-circle btn-sm"><i class="fa fa-
twitter"></i></button>

</div>

</div>

<div>    <!-- scripts -->

```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js"
```

```
integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vlsF0jrRAoQmDCKtQBHU  
uLZ9AsSv4jD4Xa"
```

```
crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/web3@1.7.4/web3.min.js"
```

```
integrity="sha512-oMd9Re3VgJcXuZJn9DN6X7S7JUc7xLYZ2UyZ85Mm/xzaW3lwBr  
4fV2zjuu/n5jY/Of/2JOx35CTa6zvQNxb31Q=="
```

```
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

```
<script>
```

```
var myaccount=localStorage.getItem("myaddress");
```

```
window.onload=function(){
```

```
//console.log(myaccount); if(myaccount!=null){
```

```
window.location.replace('account.html');
```

```
}
```

```
}
```

```
function bhej(){
```

```
var addres=document.getElementById("address").value;
```

```
localStorage.setItem("myaddress",addres);
```

```
window.location.replace('account.html');
```

```
}
```

```
</script>
```

</body>

</html>

```
pragma solidity ^0.8.0;
```

```
contract Payment{
```

```
    event message(address indexed from ,string message ,address to,string timestamp);
```

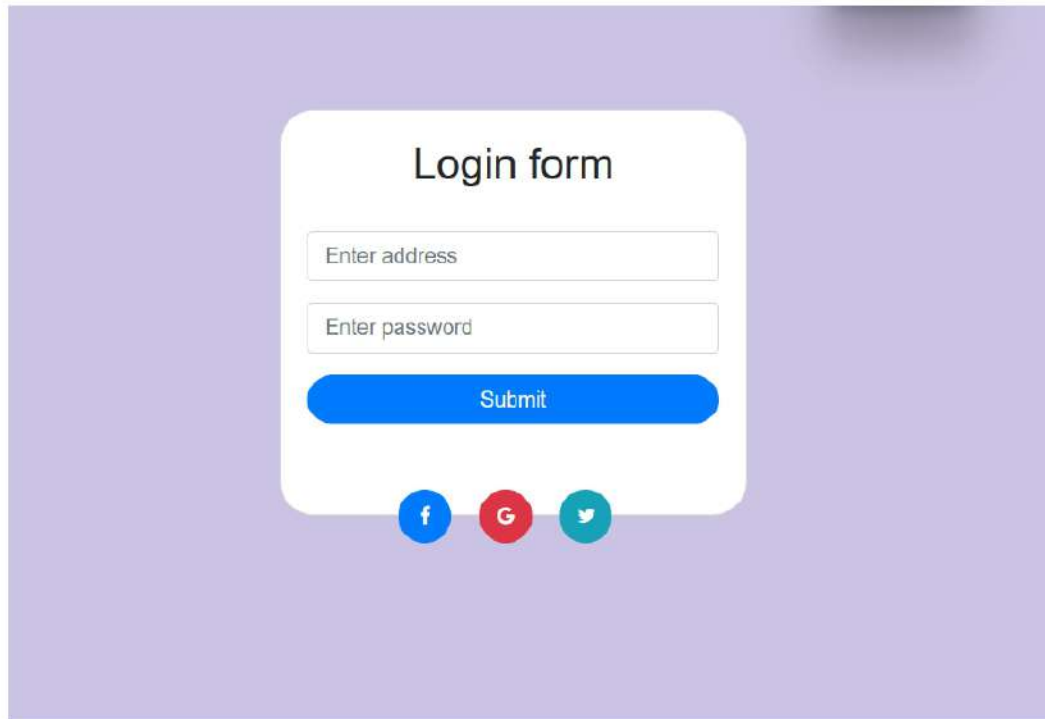
```
    function sendMessage(address _to,string memory _message,string memory time) public {
```

```
        emit message(msg.sender,_message,_to,time);
```

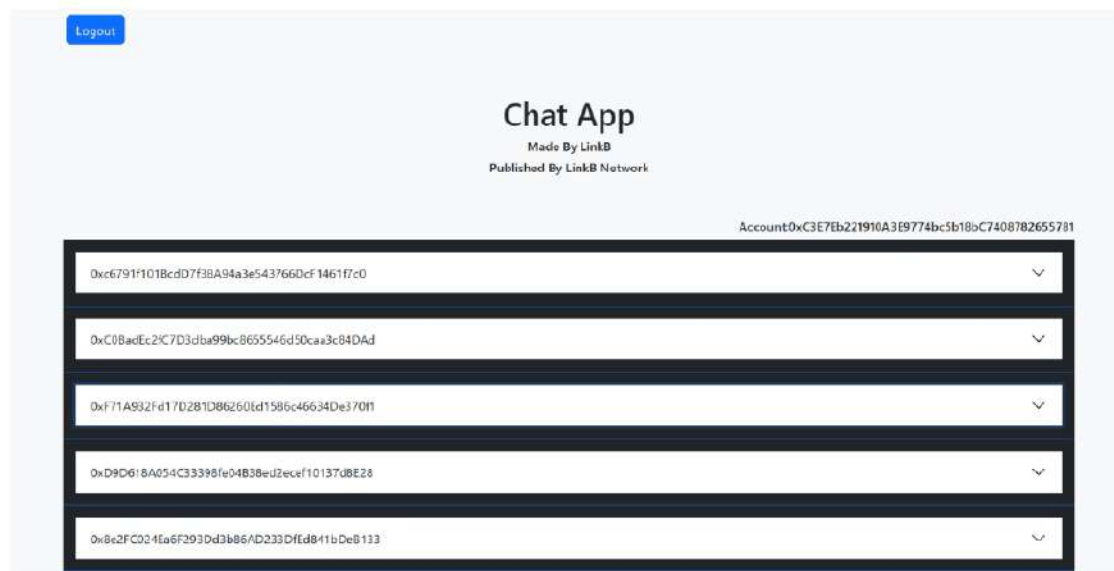
```
    }
```

```
}
```

OUTPUT



A login form titled "Login form" is centered on a light purple background. The form is a white rounded rectangle containing two input fields: "Enter address" and "Enter password". Below these fields is a blue "Submit" button. At the bottom of the form are three circular social media icons for Facebook, Google+, and Twitter.



A chat application interface titled "Chat App" is shown. It includes a "Logout" button in the top left corner. The app is credited to "Made By LinkB" and "Published By LinkB Network". The user's account ID, "Account0xC3E7Eb221910A3E9774bc5b18bC7408782655781", is displayed in the top right. The chat area contains five messages, each with a hexadecimal address and a dropdown arrow:

Message Address	Action
0xc6791f101BcdD7F3BA94a3e543766Dcf1461f7c0	▼
0xC0BadeC2C7D3dba99bc8655546d5Dcaa3c84DAd	▼
0xF71A932Fd17D281D86260Ed1585c46634De370f1	▼
0xD9D618A054C33398fe04B38ed2ecef101370BE28	▼
0x8e2FC024fa6F293Dd3b06AD233DfEd841bDe8133	▼

