**TABLE OF CONTENTS**

| **CHAPTER NO** | **TITLE** | **PAGE NO** |
| --- | --- | --- |
|  | **ABSTRACT** | **I** |
| **1.** | **INTRODUCTION** | 9  10  11  12  13  15  17 |
|  | **1.1** Background |
|  | **1.2** Objective |
|  | **1.3** Motivation |
|  | **1.4** Benefits |
|  | **1.5** Factors  **1.6** Block chain Overview |
| **2.** | **SYSTEM ANALYSIS**   * 1. Introduction   2. Proposed System   3. Problem Discussion   4. Expected Outcome   5. Overview of Existing System   6. Applications   7. Conceptual design of the project   8. Algorithm Used | 20  24  25  27  28  29  30  31 |
| **3.** | **LITERATURE REVIEW** | 33 |
|  |  |  |

| **4.** | **SYSTEM DESIGN**   * 1. Introduction   2. UML Diagrams      1. Use-case Diagram      2. Activity Diagram      3. Sequence Diagram      4. Deployment Diagram | 37  39  40  41  42 |
| --- | --- | --- |
| **5.** | **SYSTEM ARCHITECTURE** |  |
|  | **5.1** Architecture diagram | 44 |
|  |  |  |
| **6.** | **REQUIREMENTS** |  |
|  | **6.1** Software Requirement  **6.2** Hardware Requirement  **6.3** Security Considerations  **6.4** User Interface and Experience | 46  46  47  47 |

| **7.** | **SAMPLE CODE**  **7.1** Source Code | 49 |
| --- | --- | --- |
| **8.** | **OUTPUT**  **8.1**Sample Output | 69 |
| **9.** | **CONCLUSION**  **9.1** Conclusion  **9.2** Reference | 72  73 |

**Abstract**

Blockchain technology has emerged as a disruptive force in various sectors, offering decentralized and secure solutions for transactions and data management. Among the prominent blockchain platforms, Ethereum stands out with its smart contract functionality and robust ecosystem. Ethereum wallets serve as interfaces for users to interact with the Ethereum blockchain, facilitating the storage, management, and transfer of digital assets such as Ether (ETH) and tokens.This abstract explores the process of transferring funds using Ethereum wallets within the blockchain network. The transfer mechanism relies on cryptographic principles and consensus algorithms to ensure the integrity and security of transactions. Users initiate transfers by submitting signed transactions containing the recipient's address, the amount to be transferred, and any additional data required by smart contracts.The Ethereum blockchain utilizes a peer-to-peer network of nodes to validate and record transactions. Through a consensus mechanism known as Proof of Work (PoW) or the transitioning Proof of Stake (PoS), miners or validators validate blocks of transactions, adding them to the immutable blockchain ledger. This consensus mechanism ensures trust and reliability in the transfer process.

# **CHAPTER 1**

**1.INTRODUCTION:**

Blockchain technology has revolutionized the way financial transactions are conducted, offering transparency, security, and decentralization. Ethereum, one of the prominent blockchain platforms, allows users to create and execute smart contracts and decentralized applications (DApps). One of the fundamental functionalities of Ethereum is wallet-to-wallet transfers, enabling users to send and receive Ether (ETH), the native cryptocurrency of the Ethereum network. This report delves into the process of transferring Ether between Ethereum wallets using blockchain technology.

One of the fundamental aspects of utilizing Ethereum is the ability to transfer Ether between users securely and efficiently. This process, facilitated by Ethereum wallets, forms the backbone of peer-to-peer transactions within the Ethereum network. Ethereum wallets come in various forms, ranging from simple mobile applications to sophisticated hardware devices, each offering unique features tailored to different user preferences and security requirements.

In this exploration, we delve into the intricate process of Ethereum wallet transfers using blockchain technology. We will examine the underlying principles of blockchain-based transactions, elucidate the role of Ethereum wallets in facilitating these transfers, and explore the mechanisms that ensure the security and immutability of transactions on the Ethereum network.

Furthermore, we will discuss the significance of private and public keys in securing Ethereum wallets, the importance of gas fees in incentivizing miners to validate transactions, and the impact of network congestion on transaction processing times. Additionally, we will highlight emerging trends and innovations in Ethereum wallet technology, such as the integration of decentralized finance (DeFi) protocols and the adoption of layer-2 scaling solutions to enhance scalability and reduce transaction costs.

**1.1: BACKGROUND:**

Blockchain technology, pioneered by Bitcoin in 2009, has evolved into a robust and versatile platform with applications beyond cryptocurrencies. One of the most prominent blockchain platforms, Ethereum, introduced by Vitalik Buterin in 2015, enables the creation of decentralized applications (dApps) and smart contracts. Ethereum's underlying blockchain facilitates not only peer-to-peer transactions of its native cryptocurrency, Ether (ETH), but also the execution of programmable smart contracts, which automate and enforce agreements without the need for intermediaries.

Ethereum wallets serve as digital interfaces for users to interact with the Ethereum blockchain, enabling them to store, send, and receive Ether, as well as interact with smart contracts and decentralized applications. These wallets come in various forms, including software wallets (desktop, mobile, or web-based), hardware wallets (physical devices), and paper wallets (printed or written records of keys). Each wallet provides users with a unique public address, akin to an account number, and a private key, similar to a password, which grants access to the funds stored on the blockchain.

**1.2 OBJECTIVE:**

To facilitate secure and efficient Ethereum wallet transfers utilizing blockchain technology, ensuring transparent, immutable, and decentralized transactions.

**DETAILED OBJECTIVE:**

* **Secure Transactions**: Develop a system that ensures the security of Ethereum wallet transfers by employing cryptographic techniques and adhering to best practices in blockchain security protocols. Implement robust encryption methods to safeguard transaction data and prevent unauthorized access.
* **Efficient Transfer Process**: Streamline the Ethereum wallet transfer process by optimizing transaction speeds and minimizing latency. Utilize advanced consensus mechanisms and network scalability solutions to enhance transaction throughput, thereby reducing confirmation times and improving overall efficiency.
* **Blockchain Integration**: Integrate Ethereum blockchain technology into the transfer process to leverage its decentralized architecture and immutable ledger. Utilize smart contracts to automate and execute transfer instructions, eliminating the need for intermediaries and reducing transaction costs.
* **User Transparency**: Ensure transparency throughout the transfer process by providing users with real-time visibility into transaction status and confirmation. Implement user-friendly interfaces that enable stakeholders to track the progress of their transfers and access detailed transaction information.
* **Compliance and Regulation**: Incorporate compliance measures to adhere to regulatory requirements and ensure adherence to relevant laws and guidelines governing cryptocurrency transactions. Implement Know Your Customer (KYC) and Anti-Money Laundering (AML) procedures to mitigate potential risks and enhance regulatory compliance.
* **Scalability and Sustainability**: Design a scalable infrastructure capable of supporting a growing volume of Ethereum wallet transfers while maintaining optimal performance and reliability. Implement solutions to address scalability challenges and ensure long-term sustainability of the transfer system.
* **Continuous Improvement**: Continuously monitor and evaluate the Ethereum wallet transfer process to identify areas for improvement and implement enhancements accordingly. Stay abreast of emerging technologies and industry developments to adapt the transfer system to evolving user needs and market trends.

By achieving these objectives, the Ethereum wallet transfer system will provide users with a secure, efficient, and transparent means of transferring funds, thereby facilitating broader adoption of blockchain technology in financial transactions.

**1.3 MOTIVATION:**

Ethereum operates on a decentralized network of nodes, eliminating the need for central authorities or intermediaries to facilitate transactions. Ethereum wallet transfers leverage this decentralized infrastructure, enabling users to exert greater control over their funds without relying on traditional financial institutions. This decentralization aspect fosters trust and transparency in transactions, as they are recorded on an immutable ledger accessible to all network participants.

Security concerns are paramount when it comes to managing digital assets. Ethereum wallets utilize cryptographic techniques to secure private keys, which are required to access and authorize transactions. By storing Ether in Ethereum wallets, users can mitigate risks associated with centralized exchanges or traditional banking systems, which are vulnerable to hacking attacks and security breaches. The blockchain's immutable nature ensures that once a transaction is confirmed, it cannot be altered or tampered with, providing a high level of security.

Ethereum wallet transfers empower individuals with full ownership and control over their funds. Unlike traditional banking systems where account access can be restricted or frozen by third-party entities, Ethereum wallets give users the autonomy to manage their assets independently. Whether it's sending Ether to another wallet address, participating in decentralized finance (DeFi) protocols, or interacting with smart contracts, users have the freedom to transact at their discretion without seeking permission from intermediaries.

**1.4 BENEFITS:**

**Decentralization**: Ethereum operates on a decentralized network, meaning there's no central authority controlling it. Transactions occur directly between users, eliminating the need for intermediaries like banks. This decentralization fosters trust and transparency as transactions are verified by a network of nodes rather than a single entity.

**Security**: Blockchain technology employs cryptographic techniques to secure transactions. When you transfer Ethereum using a blockchain wallet, your transaction is cryptographically signed, making it extremely difficult for unauthorized parties to alter or tamper with the transaction data. Additionally, the distributed nature of blockchain ensures that there's no single point of failure, enhancing security.

**Immutability**: Once a transaction is recorded on the Ethereum blockchain, it becomes immutable, meaning it cannot be altered or deleted. This feature ensures the integrity of the transaction history and provides a high level of assurance for users.

**Fast Transactions**: Ethereum blockchain enables fast transactions compared to traditional banking systems, especially for international transfers. Transactions can be processed within minutes, regardless of the geographical location of the sender and receiver. This speed is particularly beneficial for businesses and individuals who require quick access to funds.

**Low Transaction Fees**: Ethereum transactions typically involve lower fees compared to traditional financial intermediaries like banks.This makes Ethereum an attractive option for microtransactions and cross-border payments.

**1.5 FACTORS:**

**Blockchain Technology:**

Ethereum operates on blockchain technology, a decentralized, distributed ledger system. This means that transactions are recorded across a network of computers (nodes), and each transaction is cryptographically secured.

The Ethereum blockchain is specifically designed to execute smart contracts, which are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. Smart contracts facilitate transactions automatically when certain conditions are met.

**Ethereum Wallet:**

A wallet is a digital tool that allows users to store, send, and receive Ethereum and other cryptocurrencies. Each wallet has a unique address, which is a string of alphanumeric characters.

There are different types of wallets: software wallets (desktop, mobile, or online), hardware wallets (physical devices), and paper wallets (printed documents containing keys).

When transferring Ethereum, you need to have access to a wallet and its corresponding private key, which is used to sign transactions securely.

**Transferring Ethereum:**

To transfer Ethereum from one wallet to another, the sender initiates a transaction. They specify the recipient's wallet address, the amount of Ethereum to be sent, and possibly a transaction fee.

The transaction is broadcast to the Ethereum network, where it awaits confirmation. Miners (nodes in the network) collect and validate transactions by solving complex mathematical puzzles.

Once a transaction is validated and confirmed by a certain number of blocks (confirmations), it becomes irreversible and is recorded on the blockchain.

**Factors affecting Ethereum Transfer:**

Transaction Fees: The sender may need to pay a transaction fee to incentivize miners to include their transaction in the next block. Higher fees generally result in faster confirmation times.

Network Congestion: During periods of high activity, the Ethereum network may become congested, leading to delays in transaction processing and increased fees.

Security: It's crucial to ensure the security of both the sender's and recipient's wallets. This involves protecting private keys from unauthorized access and using reputable wallet providers.

Accuracy of Transaction Details: Sending Ethereum to the wrong address or entering incorrect transaction details can result in irreversible loss of funds. It's essential to double-check all transaction details before confirming.

Smart Contract Interactions: If the transfer involves interacting with a smart contract, such as in decentralized finance (DeFi) applications, additional factors such as gas limits and contract functionality come into play.

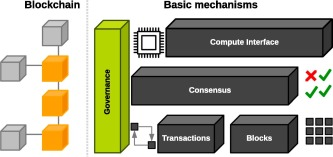
**Confirmation and Finality:**

After a transaction is included in a block and added to the blockchain, it goes through a series of confirmations as subsequent blocks are added on top of it. The number of confirmations needed for a transaction to be considered final varies but typically ranges from a few to several blocks deep.

Once a transaction has a sufficient number of confirmations, it is considered irreversible, and the transferred Ethereum becomes available in the recipient's wallet.

**1.6 BLOCKCHAIN OVERVIEW:**

In principle, a blockchain should be considered as a *distributed append-only timestamped* [*data structure*](https://www.sciencedirect.com/topics/computer-science/data-structure). Blockchains allow us to have a distributed peer-to-peer network where non-trusting members can verifiably interact with each without the need for a trusted authority ([Christidis and Devetsikiotis, 2016](https://www.sciencedirect.com/science/article/pii/S0736585318306324#b0395)). To achieve this one can consider blockchain as a set of interconnected mechanisms which provide specific features to the infrastructure, as illustrated in [Fig. 1](https://www.sciencedirect.com/science/article/pii/S0736585318306324#f0005). At the lowest level of this infrastructure, we have the signed *transactions* between peers. These transactions denote an agreement between two participants, which may involve the transfer of physical or [digital assets](https://www.sciencedirect.com/topics/computer-science/cryptographic-asset), the completion of a task, etc.

****

# 

# **CHAPTER 2**

**2.SYSTEM ANALYSIS:**

**2.1 Introduction:**

Ethereum is a decentralized blockchain platform that enables smart contracts and decentralized applications (DApps) to be built and operated without any downtime, fraud, control, or interference from third parties.

Ethereum wallets are essential tools for users to store, manage, and transfer their Ether (ETH) and other tokens on the Ethereum blockchain.

**System Components:**

**User Interface:** The frontend interface allows users to interact with the wallet application. It includes functionalities like login, balance check, transaction history, sending Ether/tokens, etc.

**Backend Server:** Handles user authentication, authorization, and business logic processing. - Communicates with Ethereum nodes to send and receive transactions.

**Blockchain Network:** Ethereum blockchain network consists of nodes that maintain the distributed ledger. - Nodes validate transactions, execute smart contracts, and maintain the state of the blockchain.

**Smart Contracts:** Smart contracts are self-executing contracts with the terms of the agreement directly written into code. - Smart contracts on Ethereum facilitate various functionalities such as token transfers, ownership verification, etc.

**Database:** Stores user information, transaction history, and other relevant data. - Ensures data persistence and accessibility.

**Functional Requirements:**

**User Registration/Login:** Users should be able to register/login to the wallet application securely.

**Balance Inquiry:** Users should be able to check their Ether and token balances.

**Transfer Funds:** Users should be able to initiate transfers of Ether and tokens to other wallet addresses.

**Transaction History:** Users should be able to view their transaction history, including details such as date, time, amount, sender/receiver addresses, and transaction status.

**Security Measures:** Implementation of security features like multi-factor authentication, encryption, and secure key management to safeguard user assets.

**Error Handling:** Proper error handling mechanisms should be in place to deal with exceptions like insufficient funds, network errors, etc.

**Non-functional Requirements:**

**Performance:** The system should handle a high volume of transactions efficiently without significant latency.

**Scalability:** The system should be designed to scale with increasing user base and transaction volume.

**Reliability:** The system should be highly available and reliable, ensuring that users can access their wallets and perform transactions at any time.

**Security:** Robust security measures should be implemented to protect user accounts, transactions, and sensitive information from unauthorized access and malicious attacks.

**Usability:** The user interface should be intuitive and user-friendly, catering to users with varying levels of technical expertise.

**Compliance:** Compliance with relevant regulations and standards related to cryptocurrency transactions and user data privacy should be ensured.

**System Interactions:**

* + User interacts with the frontend UI to perform actions like login, balance check, and fund transfer.
  + Frontend communicates with the backend server to process user requests.
  + Backend server interacts with the Ethereum blockchain network to send/receive transactions and execute smart contracts.
  + Backend server also interacts with the database to retrieve and store user data and transaction history.

**System Constraints:**

* + Dependency on Ethereum blockchain network for transaction processing.
  + Integration with third-party services for features like price feeds, gas estimation.
  + Adherence to gas fees and network congestion affecting transaction processing time and cost.

**Testing and Quality Assurance:**

* + Comprehensive testing should be conducted to ensure the correctness, reliability, and security of the system.
  + Testing methodologies include unit testing, integration testing, functional testing, security testing, and performance testing.

**Future Enhancements:**

* + Integration with decentralized finance (DeFi) protocols for additional financial services.
  + Support for hardware wallets and other secure storage solutions.
  + Implementation of advanced security features like biometric authentication, hardware security modules, etc.
  + By conducting a thorough system analysis as outlined above, developers can design and implement an Ethereum wallet transfer system that meets user requirements while ensuring security, reliability, and scalability.

**2.2 PROPOSED SYSTEM:**

**System Overview:**

**a. User Interface:** A user-friendly interface through which users can interact with the system. This interface allows users to initiate Ethereum token transfers, view transaction history, and manage their wallets.

**b. Wallet Management Module:** This module is responsible for creating and managing Ethereum wallets for users. It generates public-private key pairs for each user, which are used to securely store and access their Ethereum tokens.

**c. Ethereum Blockchain Network:** The backbone of the system is the Ethereum blockchain network. It provides the underlying infrastructure for recording transactions, validating them through consensus mechanisms, and maintaining a distributed ledger of Ethereum token balances.

**d. Smart Contracts:** Smart contracts are self-executing contracts with predefined rules and conditions encoded onto the Ethereum blockchain. In our system, smart contracts facilitate the transfer of Ethereum tokens between wallets. These contracts ensure the integrity and security of transactions by executing predefined logic without the need for intermediaries.

**e. Transaction Processing Engine:** This component processes transfer requests initiated by users. It interacts with the Ethereum blockchain network to execute smart contracts and update the ledger with the details of each transaction.

**2.3 PROBLEM DISCUSION:**

**a. Security Concerns:** Traditional banking systems are susceptible to hacks and fraudulent activities. Similarly, Ethereum wallets are vulnerable to hacking attempts, phishing attacks, and malware. Ensuring robust security measures is crucial to safeguard users' funds and personal information.

**b. User Experience:** Complex interfaces and lengthy transaction processes can deter users from engaging with Ethereum wallets. Enhancing user experience by simplifying the transfer process and providing intuitive interfaces is essential for widespread adoption.

**c. Transaction Speed and Scalability:** Ethereum's network congestion can lead to delays in transaction processing and higher fees during peak periods. Scalability issues have also been a concern, limiting the platform's ability to handle a large number of transactions simultaneously.

**d. Transparency and Traceability:** Transparency in transaction history and real-time tracking of transfers are vital for users to trust the system. Blockchain's inherent transparency can be leveraged to provide users with immutable records of transactions, ensuring accountability and auditability.

**e. Cross-Platform Compatibility:** Interoperability between different wallets and blockchain platforms is essential for seamless transfer of assets. Supporting multiple wallets and standardizing protocols can improve accessibility and usability for users.

**Objectives:**

* + Develop a secure and user-friendly Ethereum wallet transfer system leveraging blockchain technology.
  + Implement robust security measures to protect users' funds and personal information.
  + Enhance user experience through intuitive interfaces and streamlined transaction processes.
  + Optimize transaction speed and scalability to mitigate delays and reduce fees.
  + Ensure transparency and traceability of transactions by leveraging blockchain's immutable ledger.
  + Ensure cross-platform compatibility to enable interoperability between different wallets and blockchain networks.

**Proposed Solution:**

* + Design and develop a decentralized application (DApp) for Ethereum wallet transfer, utilizing smart contracts for secure and automated transactions.
  + Implement multi-factor authentication (MFA), encryption, and other security measures to enhance the security of the platform.
  + Utilize Ethereum's layer 2 solutions such as sidechains or scaling solutions like Optimistic Rollups to improve transaction throughput and reduce latency.
  + Incorporate user-friendly interfaces with features such as QR code scanning, transaction history, and real-time notifications.
  + Leverage blockchain's transparency to provide users with verifiable transaction records and audit trails.
  + Standardize protocols for wallet interoperability, enabling seamless transfer of assets between different platforms.

**2.4 Expected Outcome:**

* + A secure and efficient Ethereum wallet transfer system that enhances user experience and mitigates common challenges associated with blockchain transactions.
  + Improved security measures to protect users' assets and personal information from unauthorized access and fraudulent activities.
  + Enhanced transaction speed and scalability, ensuring fast and cost-effective transfers even during peak network congestion.
  + Transparent and traceable transaction records accessible to users for auditability and accountability.
  + Increased adoption of Ethereum wallets due to improved usability, interoperability, and reliability.

**2.5 Overview of Existing System:**

**Ethereum Blockchain:** Ethereum is a decentralized platform that enables developers to build and deploy smart contracts and decentralized applications (DApps). It utilizes blockchain technology to maintain a distributed ledger of transactions.

**Wallets:** Ethereum wallets are software applications that allow users to interact with the Ethereum blockchain. There are several types of wallets including desktop wallets, mobile wallets, web wallets, hardware wallets, and paper wallets. Each wallet has its own unique features and security measures.

**Public and Private Keys:** Ethereum wallets use public and private keys for secure authentication and transaction signing. Public keys are used to generate wallet addresses, which are alphanumeric strings representing destinations for Ethereum transactions. Private keys are kept secret and are used to sign transactions to prove ownership of the wallet.

**Transactions:** Ethereum transactions involve sending Ether (ETH), the native cryptocurrency of the Ethereum network, or interacting with smart contracts deployed on the Ethereum blockchain. Transactions are broadcasted to the network and included in blocks by miners.

**Gas:** Ethereum transactions require a fee known as "gas" to be paid in order to incentivize miners to include the transaction in a block. Gas prices vary depending on network congestion and the complexity of the transaction.

**Blockchain Explorer:** Users can explore Ethereum transactions and wallet balances using blockchain explorers, which are web-based tools that provide a graphical interface to interact with the Ethereum blockchain.

**Smart Contracts:** Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They enable trustless and transparent transactions on the Ethereum blockchain without the need for intermediaries.

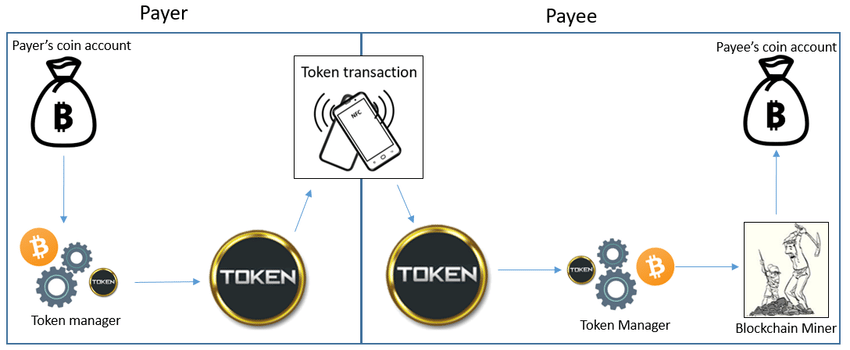
**Security:** Security is of utmost importance in the Ethereum ecosystem. Users must safeguard their private keys and use reputable wallet providers to prevent unauthorized access to their funds. Additionally, smart contracts must be thoroughly audited to mitigate the risk of vulnerabilities and exploits.

**Interoperability:** Ethereum is interoperable with other blockchains through technologies such as bridges and interoperability protocols. This enables the transfer of assets and data between different blockchain networks, expanding the functionality and utility of the Ethereum ecosystem.

**2.6 Applications:**

* **Peer-to-Peer Transactions**
* **Smart Contracts**
* **Decentralized Finance (DeFi)**
* **Tokenization of Assets**
* **Initial Coin Offerings (ICOs) and Token Sales**
* **Gaming and Non-Fungible Tokens (NFTs)**
* **Supply Chain Management**
* **Identity Management**

**2.7 ConceptualDesign:**



**2.8 ALGORITHM USED:-**

Ethereum wallets are designed to hold cryptographic keys instead of directly storing ether or tokens1. The keys are generated using cryptographic algorithms. Here are the key algorithms involved:

Elliptic Curve Digital Signature Algorithm (ECDSA): This is used for transaction verification and wallet security in Ethereum2. It provides a secure and efficient way to sign and verify digital signatures, ensuring the integrity and authenticity of transactions on the blockchain2.

Keccak-256 cryptographic hash function: Ethereum uses this in a consensus engine known as Ethash3. It is a version of Secure Hash Algorithm Version 3 or SHA-33. It can be used for transparency, scam-less authentication, encryption, and pseudo-random number generation3.

Ethash: This is the Ethereum algorithm that underpins Ethereum’s proof of work mining process4. It is an altered version of the previous proof of work algorithm, Dagger-Hashimoto4.

secp256k1: This is the elliptic curve used in the implementation of Ethereum

# **CHAPTER 3**

**3.LITERATURE REVIEW:**

| Sr no | Paper title | General idea | Advantages & limitations |
| --- | --- | --- | --- |
| 1 | Blockchain and Cryptocurrencies: Model, Techniques, and Applications [2018] | A survey of current cryptocurrencies to understand blockchain & its different types. | Advantages: Provides different incentive models, ecosystem & applications of the blockchain. Explains blockchain in a layered architecture. Limitations: Does not provide any solid architecture for its stated application. |
| 2 | A Brief Survey of Cryptocurrency Systems[2017] | It evaluates the strengths, weaknesses, and possible threats to all major mining strategy. It outlines how Cryptocurrencies mine, where they have comparable performance and assurance, and where they have unique threats and strengths. | Advantages: 1. Currently, major Cryptocurrencies use Proof of Work, Proof of Stake or a combination of the both for mining. 2. A combination of the both is found to be effective. 3. Typically memory-intensive hash functions have been found to be faster mining algorithms. Limitations: A majority of hash algorithms are CPU-intensive and the others are memory intensive. 2. While Proof of Work is resource intensive, Proof of Stake cannot act independently. 3. Cryptocurrencies are still experimenting with their mining protocols and algorithms to optimize their performance. No full proof algorithm has been found yet. |
| 3 | Blockchain: Future of Financial and Cyber Security[2016] | This paper explains the concept, characteristics, need for Blockchain and how Bitcoin works. It attempts to highlights role of Blockchain in shaping the future of banking. | Advantages: 1.The decrease in device cost 2.Increases computing power  Limitations: 1. If an attack was done by an attacker then there will be a loss of all bitcoins, we can't recover it because the government is not involved in |
| 4 | Bitcoin: A Peer-to-Peer Electronic Cash System[2008] | A distributed peer to peer system working under the blockchain framework | Advantages: Cryptocurrency without any central authority. Successful POW mechanism. Limitations: The cost of POW consensus protocol will keep increasing as more people join the network. |
| 5 | Trust Your Wallet : a New Online Wallet Architecture for Bitcoin [2017] | It introduces a wallet which is highly secured by Multiple signatures. | Advantages: The scalability of disaster recovery center Limitations: If we lost one of the keys then we are not able to recover that key |
| 6 | A survey on the security of blockchain systems [2017] | Detail survey of the security issues in current systems and existing solutions | Advantages: A careful comparison between bitcoin and ethereum. Different aspects of system vulnerability Limitations: Cryptocurrency will need more methods to achieve security and privacy. |

# **CHAPTER 4**

**4.SYSTEM DESIGN:**

**4.1 Introduction:**

The objective of this project to create a de-centralized blockchain based cryptocurrency and an e-wallet to access currency. The project entails creating a generalized blockchain API which can later be used for further development in the blockchain field of applications. This blockchain API will be used to construct a cryptocurrency from scratch. Implement a new hashing function for the blockchain which will take data and create a fixed length output. Implementing a highly secure and personal e-wallet system to access and control the said cryptocurrency. This e-wallet system will let the user control and transact the currency efficiently. It provides the user with full authority over the token currencies. The aim is also to make the currency secure, easy to access, fast and as cheap to avail as possible.

1. Creating a crypto-currency with proof-of-authority but without a central point of failure.

2. Generalized public blockchain API.

3. A hybrid consensus protocol for blockchain.

4. A secure and high availability e-wallet for easy to facilitate transactions.

**4.2 UML Diagram:**

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

▪ activities

▪ actors

▪ business processes

▪ database schemas

▪ (logical) components

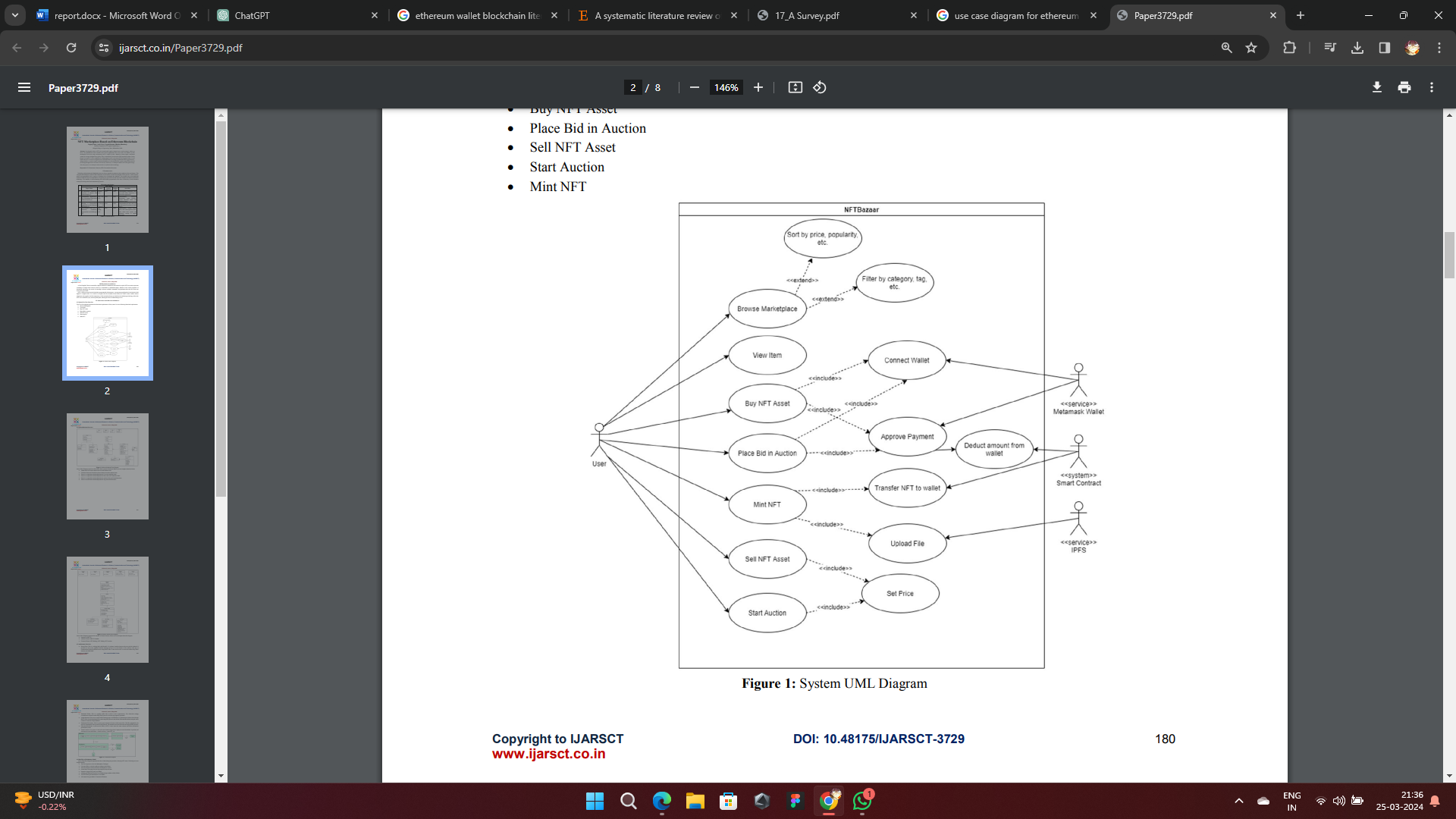
▪ programming language statements

▪ Reusable software components.

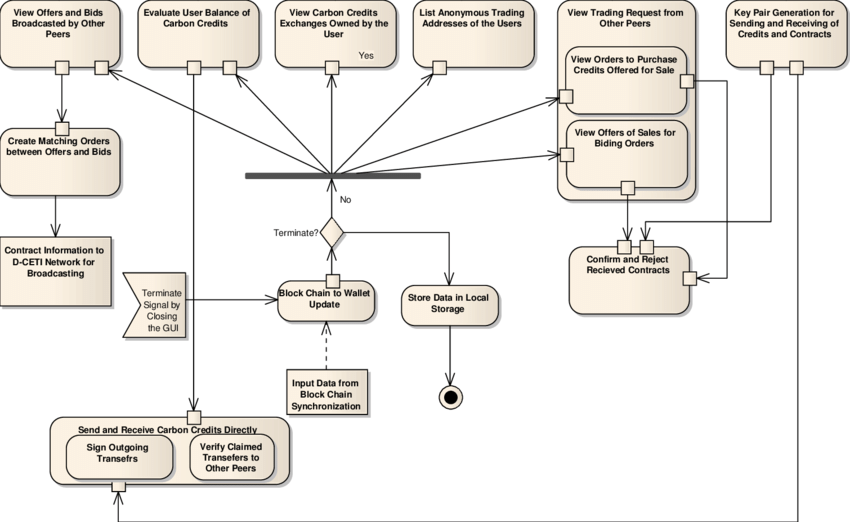
UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementationtechnologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language.UML aims to be a standard modeling language which can model concurrent and distributed systems.

**4.2.1 USE CASE:**

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. Actors must be able to make decisions, but need not be human: "An actor might be a person, a company or organization, a computer program or a computer system — hardware, software, or both

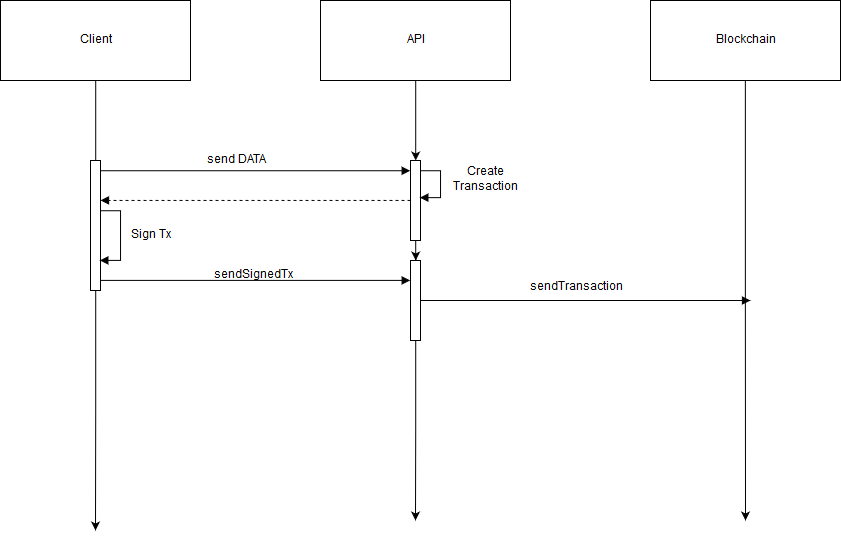


**4.2.2 ACTIVITY DIAGRAM** A diagram of the sequence of movements or actions of people or things involved in a complex system or activity and a graphical representation of a computer program in relation to its sequence of functions (as distinct from the data it processes).



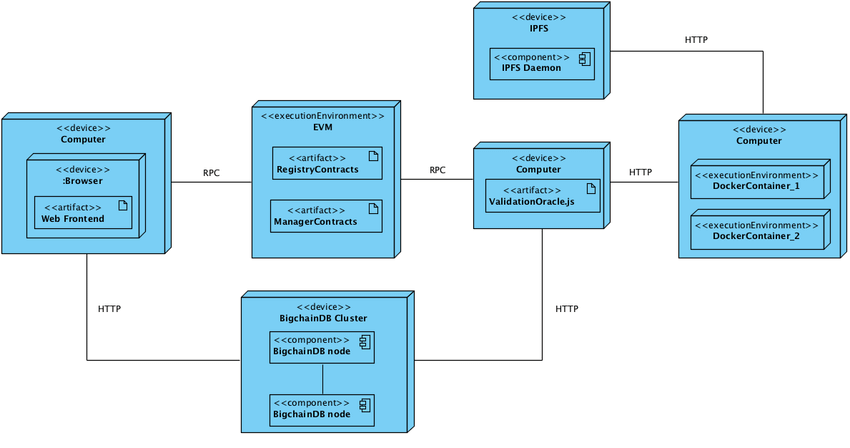
**4.2.3 SEQUENCE DIAGRAM**

A sequence diagram or system sequence diagram shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the process needed to carry out the functionality



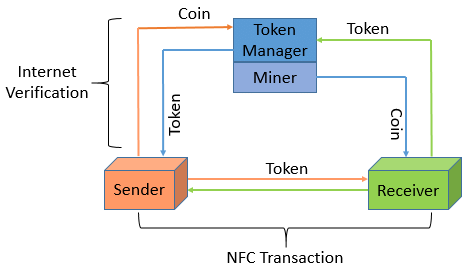
**4.2.4 DEPLOYMENT DIAGRAM**

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components exist, what software components run on each node, and how the different pieces are connected



# **CHAPTER 5**

**5. SYSTEM ARCHITECTURE:**



# **CHAPTER 6**

**6.SYSTEM REQURIMENTS:**

1. **Software Requirements**:
   * Operating System: Most commonly used operating systems like Windows 10,11.
   * Development Environment: You would need software development tools such as IDEs (Integrated Development Environments) like Visual Studio Code, Sublime Text, or Atom.
   * Programming Languages: JavaScript, HTML, CSS, JSON for building the backend and frontend components of your wallet.
   * Libraries and Frameworks: You might use libraries and frameworks like Web3.js, Ethereum.js for easier interaction with Ethereum.
2. **Hardware Requirements**:
   * Processor: A modern multi-core processor (e.g., Intel Core i5 or AMD Ryzen) would be suitable for development purposes.
   * RAM: At least 4GB of RAM is recommended for running Ethereum client software and development tools comfortably.
   * Storage: You'll need sufficient disk space for storing the Ethereum blockchain data if you plan to run a full node. Additionally, storage for your development tools and project files is necessary.
   * Internet Connection: A stable internet connection is required for syncing with the Ethereum network and for interacting with smart contracts and DApps.
3. **Security Considerations**:
   * Secure Development Practices: Follow secure coding practices to minimize vulnerabilities in your wallet application.
   * Data Encryption: Implement encryption mechanisms to protect sensitive data such as private keys.
   * Backup and Recovery: Include features for users to easily backup their wallets and recover access in case of loss or theft.
   * Compliance: Ensure compliance with relevant regulations and standards, especially regarding user data protection and financial transactions.
4. **User Interface and Experience**:
   * Design: Create an intuitive and user-friendly interface for your Ethereum wallet application.
   * Cross-Platform Compatibility: Consider developing desktop, web, and mobile versions of your wallet to cater to a broader audience.
   * Accessibility: Ensure accessibility features are implemented for users with disabilities.
   * Testing: Thoroughly test your wallet application across different devices and platforms to ensure consistency and reliability.

# **CHAPTER 7**

**7. SAMPLE CODE:**

**HTML Code**:

<html>

<!-- HEAD -->

<head>

<!-- TITLE -->

<title>Progressive Web Wallet</title>

<!-- META TAGS -->

<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">

<meta name="theme-color" content="#000000">

<!-- MANIFEST -->

<link rel="manifest" href="manifest.json">

<!-- ICON -->

<link rel="icon" type="image/png" href="img/icon.png">

<!-- STYLESHEETS -->

<link href="css/style.css" type="text/css" rel="stylesheet">

<link href="css/spinner.css" type="text/css" rel="stylesheet">

<!-- FONTS -->

<link href="https://fonts.googleapis.com/css?family=Ubuntu" rel="stylesheet">

<!-- ANALYTICS -->

<script async src="https://www.googletagmanager.com/gtag/js?id=UA-110452092-1"></script>

<script>

window.dataLayer = window.dataLayer || [];

function gtag(){dataLayer.push(arguments);}

gtag('js', new Date());

gtag('config', 'UA-110452092-1');

</script>

</head>

<!-- BODY -->

<body>

<!-- PAGE: NEW WALLET -->

<div class="content" id="content-new-wallet" hidden>

<!-- INPUT: PASSWORD -->

<div>Enter a password for your wallet<br><br></div>

<input type="password" id="input-new-password"></input>

<!-- BUTTON: GENERATE WALLET -->

<div class="button" onclick="generateWallet()">GENERATE WALLET</div>

</div>

<!-- PAGE: MAIN -->

<div class="content" id="content-main" hidden>

<!-- QR CODE -->

<img id="qr">

<!-- ADDRESS -->

<a id="etherscan-address" href="#" target="\_blank"><div id="address"></div></a>

<!-- COPY TO CLIPBOARD -->

<div class="clipboard" id="clipboard-address"><img src="img/clipboard.svg" width=20></div>

<!-- BALANACE -->

<div id="balance"><span id="balance-eth">0</span> ETH <br>(<span id="balance-base">0</span> <span id="base-currency">INR</span>)</div>

<!-- BUTTON: SEND -->

<div class="button" id="button-send-ether" onclick="setPage('send')">SEND ETHER</div>

<!-- BUTTON: SETTINGS -->

<div class="button" id="button-delete" onclick="setPage('settings')">SETTINGS</div>

</div>

<!-- PAGE: SEND -->

<div class="content" id="content-send" hidden>

<!-- INPUT: ADDRESS -->

<div class="input-label">Address</div>

<input type="text" id="input-send-address" placeholder="0x0"></input>

<!-- INPUT: AMOUNT -->

<div class="input-label">Amount</div>

<input type="number" id="input-send-amount" placeholder="0.0" min="0.0" step="0.1"></input>

<!-- BUTTON: SEND -->

<div class="button" onclick="sendEther()">SEND</div>

<!-- BUTTON: SCAN -->

<div class="button" onclick="launchScanner()">SCAN QR CODE</div>

<!-- BUTTON: CANCEL -->

<div class="button" onclick="clearSendInputs(); setPage('main')">CANCEL</div>

</div>

<!-- PAGE: TRANSACTION -->

<div class="content" id="content-tx" hidden>

<div>Your transaction has been broadcast to the network</div>

<!-- BUTTON: VIEW TRANSACTION -->

<a href="#" id="etherscan-tx" target="\_blank"><div class="button">VIEW TRANSACTION</div></a>

<!-- BUTTON: RETURN TO WALLET -->

<div class="button" onclick="setPage('main')">RETURN</div>

</div>

<!-- PAGE: TRANSACTION -->

<div class="content" id="content-funding" hidden>

<div>1 ETH has been requested from MetaMask's Ropsten faucet</div>

<!-- BUTTON: VIEW TRANSACTION -->

<a href="#" id="etherscan-tx-funding" target="\_blank"><div class="button" id="button-view-transaction">VIEW TRANSACTION</div></a>

<!-- BUTTON: RETURN TO WALLET -->

<div class="button" onclick="setPage('main')">RETURN</div>

</div>

<!-- PAGE: QR SCANNER -->

<div class="content" id="content-scanner" hidden>

<!-- VIDEO: QR SCANNER PREVIEW -->

<div id="video-container">

<video id="video-scanner"></video>

</div>

<!-- BUTTON: SWITCH CAMERA -->

<div class="button" onclick="switchCamera()">SWITCH CAMERA</div>

<!-- BUTTON: RETURN TO WALLET -->

<div class="button" onclick="stopScanner()">CANCEL</div>

</div>

<!-- PAGE: SETTINGS -->

<div class="content" id="content-settings" hidden>

<!-- SELECTOR: BASE CURRENCY -->

<div>

Base currency:

<select id="select-base-currency" class="dropdown" onchange="selectBaseCurrency()">

<option value="USD">USD</option>

<option value="EUR">EUR</option>

<option value="INR">INR</option>

</select>

</div>

<!-- BUTTON: REVEAL PRIVATE KEY -->

<div class="button" onclick="revealPrivateKey()">REVEAL PRIVATE KEY</div>

<!-- BUTTON: DELETE WALLET -->

<div class="button" onclick="deleteWallet()">DELETE WALLET</div>

<!-- BUTTON: RETURN TO WALLET -->

<div class="button" onclick="setPage('main')">RETURN</div>

</div>

<!-- PAGE: PRIVATE KEY -->

<div class="content" id="content-private-key" hidden>

<!-- PRIVATE KEY -->

<div id="private-key"></div>

<!-- COPY TO CLIPBOARD -->

<div class="clipboard" id="clipboard-private-key"><img src="img/clipboard.svg" width=20></div>

<!-- BUTTON: RETURN TO WALLET -->

<div class="button" onclick="hidePrivateKey(); setPage('main');">RETURN</div>

</div>

<!-- SPINNER -->

<div id="spinner">

<div id="spinner-container">

<img id="spinner-image" src="img/ethereum\_logo\_transparent\_background.png">

</div>

</div>

</body>

<script src="index.js"></script>

<script>

window.init()

</script>

</html>

**CSS Code:-**

body {

background-image: url("../img/zigzag.png");

background-repeat: repeat;

color: white;

font-family: "Courier New";

overflow-x: hidden;

}

.content {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

text-align: center;

width: 100%;

/\*border: 1px solid white;\*/

}

#header {

position: absolute;

top: 40px;

font-family: "Ubuntu";

font-weight: bold;

text-align: center;

width: 100%;

font-size: 36px;

}

#qr {

text-align: center;

border-radius: 10px;

}

.clipboard {

cursor: pointer;

}

#clipboard-address {

position: relative;

top: -56px;

left: 134px;

}

#clipboard-private-key {

margin-top: 20px;

}

#private-key {

position: relative;

word-wrap: break-word;

left: 50%;

transform: translate(-50%, 0);

width: 310px;

}

#address {

margin-top: 20px;

font-size: 12px;

}

#balance {

/\*margin-top: 20px;\*/

font-size: 24px;

}

input {

font-family: "Courier New";

background: transparent;

color: white;

border: 1px solid grey;

}

input:focus {

outline: none;

}

.input-label {

text-align: center;

margin-top: 20px;

}

#input-send-address {

width: 290px;

}

#input-send-amount {

font-size: 18px;

width: 100px;

text-align: center;

}

.button {

border: 1px solid white;

background: white;

color: black;

margin-top: 20px;

font-family: "Ubuntu";

font-weight: bold;

padding: 5px;

border-radius: 5px;

width: 200px;

position: relative;

left: 50%;

transform: translate(-50%, 0);

user-select: none;

cursor: pointer;

}

.button a {

color: black;

text-decoration: none;

}

a {

color: white;

text-decoration: none;

}

.dropdown {

font-family: "Courier New";

font-size: 16px;

border: none;

outline: none;

margin-top: 20px;

background: white;

color: black;

border-radius: 5px;

}

#video-scanner {

position: relative;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 250px;

}

#video-container {

background: black;

position: relative;

left: 50%;

transform: translate(-50%, 0);

height: 250px;

width: 250px;

overflow: hidden;

border-radius: 5px;

}

**Javascript Code:-**

const wallet = require('./wallet.js')

const ui = require('./ui.js')

const spinner = require('./spinner.js')

const tx = require('./tx.js')

const faucet = require('./faucet.js')

const scanner = require('./scanner.js')

const clipboard = require('./clipboard.js')

const analytics = require('./analytics.js')

const webworkify = require('webworkify')

const worker = webworkify(require('./worker.js'))

/\* GENERAL \*/

window.init = function() {

// If no base currency set -> set base currency

if (!localStorage.getItem("baseCurrency")) {

localStorage.setItem("baseCurrency", "USD")

}

// If wallet already created -> go to "main" page

if (localStorage.getItem("address") && localStorage.getItem("encryptedKey")) {

window.setPage("main")

// Else -> go to "new wallet" page

} else {

window.setPage("new-wallet")

}

// Stop spinner

spinner.stop()

// Send analytics

analytics.sendEvent('app', 'launch')

}

window.setPage = function(page) {

// Hide all content divs

elements = Array.from(document.getElementsByClassName("content"))

elements.map((el) => hide(el))

// Show selected div

show(document.getElementById("content-" + page))

if (page === "main") {

ui.refresh()

}

}

/\* GENERATE WALLET \*/

window.generateWallet = function() {

// Get password

var password = document.getElementById("input-new-password").value

// Start spinner

spinner.start()

// Add web worker event listener

worker.addEventListener('message', (e) => {

// Remove web worker event listener

worker.removeEventListener('message', this)

// Store encrypted private key and address in localStorage

localStorage.setItem("encryptedKey", e.data.encryptedKey)

localStorage.setItem("address", e.data.address)

// Go to "main" page

setPage('main')

// Stop spinner

spinner.stop()

// Send analytics event

analytics.sendEvent('wallet', 'generate')

})

// Post password to address generating web worker

worker.postMessage({command: "generate", password: password})

}

/\* GET ETHER \*/

window.getEther = function() {

// Start spinner

spinner.start()

// Get address

let address = localStorage.getItem("address")

// Request ether from the metamask ropsten faucet

faucet.receive(address)

.then((txHash) => {

// Log TX hash to console

console.log("Faucet Tx Hash: " + txHash)

// Update "view transaction" button link

document.getElementById("etherscan-tx-funding").href = "https://ropsten.etherscan.io/tx/" + txHash

// Go to funding page

setPage("funding")

// Stop spinner

spinner.stop()

// Send analytics event

analytics.sendEvent('wallet', 'faucet')

})

.catch((err) => {

alert(err)

spinner.stop()

})

}

/\* SEND ETHER \*/

window.sendEther = function() {

// Prompt user for password

let password = prompt("Enter password")

// Fetch encrypted key from localStorage

let encryptedKey = localStorage.getItem("encryptedKey")

// Decrypt private key

let privateKey = wallet.decryptPrivateKey(encryptedKey, password)

// If correct password entered -> perform transaction

if (privateKey) {

// Start spinner

spinner.start()

// Get address and amount from input fields

let to = document.getElementById("input-send-address").value

let amount = document.getElementById("input-send-amount").value

// Send

tx.send(to, amount, privateKey)

.then((txHash) => {

// Log transaction hash to console

console.log("Tx Hash:" + txHash)

// Update UI elements

document.getElementById("etherscan-tx").href = "https://ropsten.etherscan.io/tx/" + txHash

// Set page

setPage("tx")

// Stop spinner

spinner.stop()

// Send analytics event

analytics.sendEvent('wallet', 'tx')

})

.catch((err) => {

spinner.stop()

alert(err)

})

// Else alert user of incorrect password

} else {

alert('Wrong password')

}

}

window.clearSendInputs = function() {

clear(document.getElementById('input-send-address'))

clear(document.getElementById('input-send-amount'))

}

window.launchScanner = function() {

window.setPage("scanner")

scanner.scan()

.then((content) => {

ui.updateSendForm(content)

window.setPage("send")

})

}

window.stopScanner = function() {

scanner.stop()

window.setPage("send")

}

window.switchCamera = function() {

scanner.switchCamera()

}

/\* SETTINGS \*/

window.deleteWallet = function() {

let confirmation = confirm("Are you sure you want to delete your wallet?")

if (confirmation) {

wallet.delete()

setPage('new-wallet')

// Send analytics event

analytics.sendEvent('wallet', 'delete')

}

}

window.selectBaseCurrency = function() {

let selector = document.getElementById('select-base-currency')

let index = selector.selectedIndex

let options = selector.options

let currency = options[index].value

localStorage.setItem('baseCurrency', currency)

ui.refresh()

}

window.revealPrivateKey = function() {

let password = prompt("Enter password")

let encryptedKey = localStorage.getItem('encryptedKey')

let privateKey = wallet.decryptPrivateKey(encryptedKey, password)

if (privateKey) {

document.getElementById('private-key').innerHTML = privateKey

document.getElementById('clipboard-private-key').setAttribute('data-clipboard-text', privateKey)

setPage('private-key')

} else {

alert("Invalid password")

}

}

window.hidePrivateKey = function() {

document.getElementById('private-key').innerHTML = ""

document.getElementById('clipboard-private-key').setAttribute('data-clipboard-text', "")

}

/\* PRIVATE FUNCTIONS \*/

var hide = function(el) {

el.setAttribute("hidden", true)

}

var show = function(el) {

el.removeAttribute("hidden")

}

var clear = function(el) {

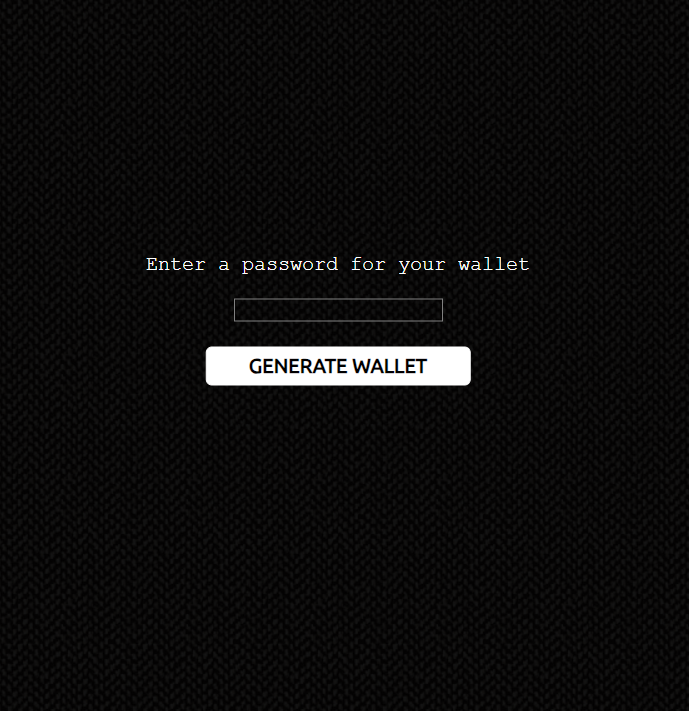
el.value = ""

}

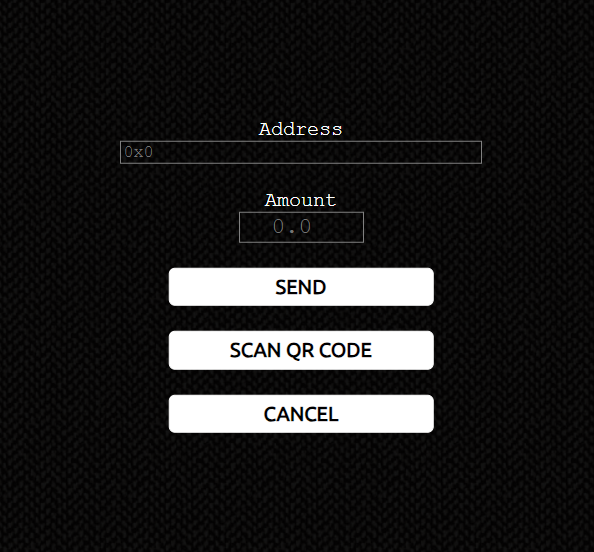
# **CHAPTER 8**

**8.SAMPLE OUTPUT:**

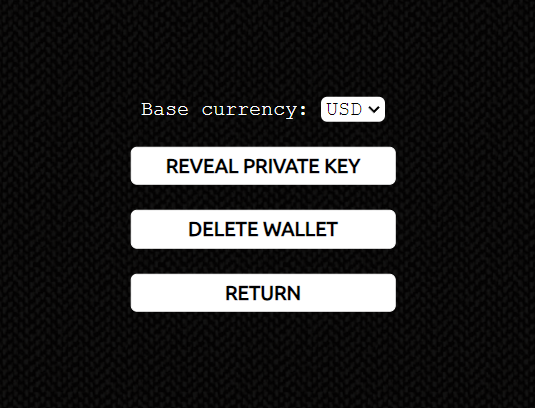
**Enter password to generate wallet:**

****

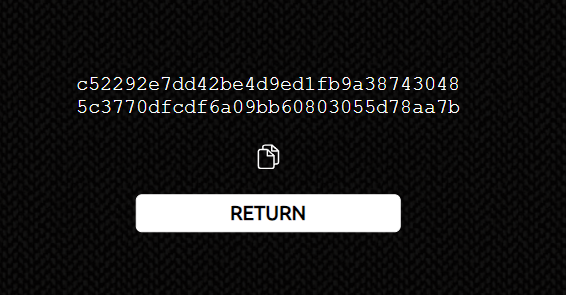
**Enter address:**



**Settings:**



**Reveal private key:**



# **CHAPTER 9**

**CONCLUSION:**

Implementing an Ethereum wallet utilizing blockchain technology offers a robust solution for securely storing, managing, and transacting with digital assets. The conclusion of such an endeavor involves summarizing the key benefits, challenges, and future prospects of Ethereum wallets in the context of blockchain technology.

Ethereum wallets represent a pivotal component of the blockchain ecosystem, offering users secure, decentralized, and interoperable solutions for managing their digital assets. While challenges remain, ongoing advancements in technology and regulation position Ethereum wallets for continued growth and innovation in the years to come.

Ethereum wallets represent more than just a tool for managing digital assets—they embody the principles of decentralization, security, and innovation that underpin the blockchain revolution. By addressing challenges, engaging communities, promoting user education, and embracing collaboration, Ethereum wallet providers can realize the full potential of this transformative technology and empower individuals worldwide to participate in the decentralized economy of the future.

Ethereum's vibrant community of developers, enthusiasts, and users plays a crucial role in the evolution of Ethereum wallets. Community-driven initiatives, such as open-source wallet development, contribute to innovation and collaboration within the ecosystem. Additionally, community feedback and engagement provide valuable insights for wallet providers to improve user experience and address emerging needs.

While Ethereum faces scalability challenges due to network congestion and high gas fees, ongoing development efforts, such as Ethereum 2.0, aim to improve scalability through solutions like sharding and proof-of-stake consensus. These enhancements promise to enhance the user experience and reduce transaction costs associated with Ethereum wallets.

**REFERENCES**:

[1] D. Chaum, “Untraceable electronic main, return addresses, and digital

pseudonyms,” in Communications of the ACM, vol. 24, no. 2, pp. 84-88,

February 1981

[2] L. Law, S. Sabett, and J. Solinas, “How to make a mint: the

cryptography of anonymous electronic cash,” American University Law

Review, vol. 46, no. 4, pp. 1131-1162, 1996

[3] C. Dwork and M. Naor, “Pricing via processing or combatting junk

mail,” in 12th Annual International Cryptology Conference, pp. 139-

147, 1992

[4] W. Dai, “B-money,” 1998, available at: http://www:weidai:com/bmoney

[5] H. Finney, “RPOW,” 2004, available at:

http://nakamotoinstitute.org/finney/rpow/

[6] N. Szabo, “Bit Gold,” 2005, available at:

http://unenumerated.blogspot.rs/2005/12/bit-gold.html

[7] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: a technical

survey on decentralized digital currencies,” IEEE Communications

Surveys & Tutorials, vol. 18, no. 3, pp. 2084-2123, March 2016

[8] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, “KARMA: A

secure economic framework for peer-to-peer resourse sharing,” 1st

Workshop on Economics of Peer-To-Peer Systems, 2003

[9] N. Szabo, “Secure property titles with owner authority,” 1998, available

at: http://nakamotoinstitute.org/secure-property-titles/