Design and Implementation of a Blockchain-Based Auction Bidding System

By

Ahmed Elayyan, Omar Abushaqra & Mohammed Abu Laila

Supervised by

Dr. Anastassia Gharib

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE
in

Networks and Information Security Engineering

at

PRINCESS SUMAYA UNIVERSITY FOR TECHNOLOGY

Amman, Jordan

Second Semester 2023 -2024

This is to certify that I have examined

this copy of an engineering documentation by

Ahmed Elayyan & Omar Abushaqra & Mohammed Abu Laila

And have found that it is complete and satisfactory in all respect,
And that any and all revisions required by the final Examining Committee have been made

Dr. Anastassia Gharib

Abstract

This project designs and implements a decentralized online auction system using blockchain technology to secure records, authenticate, and manage user bids on the platform. The selected auction type is a sealed auction for single-item and multi-item sales. Sellers are able to sell their products in a forward auction. Moreover, the website adopts single-sided auctions of participation, allowing users to browse and bid. New users can sign up after entering their credentials and location. Our authentication system contains a username and password to authenticate users. This ensures that only authorized users can access the platform and participate in the auction. Also, the website encourages customer engagement in the bidding process by offering an incentive lottery to win prizes.

Table of Contents

List of Figures

List of Tables

1 Introduction

The Internet has transformed many aspects of everyday life in recent years, from communication to transactions and services. Electronic auctions, representing an evolution from traditional methods, merge Internet technology and auction mechanisms designed to save time and exceed geographical limitations. These systems have gained massive popularity in e-commerce, offering accessibility globally. An auction is a structured process where items are offered for sale through competitive bidding, with participants placing bids, and the items going to the highest bidder.

Electronic bidding involves online platforms facilitating such auctions, allowing participants to participate from anywhere with an internet connection. These platforms are created through websites. To create a website, developers and associated parties select an appropriate technology stack based on the project's requirements. Common stacks include front-end, back-end, database, and more. An example of an electronic bidding platform is eBay [16]. Advantages of electronic bidding include increased accessibility, faster transactions, and a global reach. However, there are multiple challenges in embedding auction websites, such as challenges related to security concerns, potential fraud, and the establishment of user trust.

Meanwhile, blockchain is a new technology. It records transactions through a linked chain of data blocks using a secure decentralized digital ledger. It transformed this landscape by recording all auction-related data through its transactions, creating a decentralized ledger that ensures transparency and integrity.

Ethereum smart contract technology emerges as a solution by integrating blockchain's decentralized nature, potentially eliminating the intermediary role in electronic bidding, and facilitat ing direct transactions between sellers and buyers. These smart contracts automate bidding processes, verification, and transactions without mediators. By applying the terms of the auction agreements, smart contracts on the Ethereum blockchain enhance security, transparency, and trust. This can revolutionize the auction, providing a decentralized and secure environment that minimizes fraud risks and eliminates the need for third-party.

In the following, we will introduce the motivation for the work, project overview and objectives, some selected design requirements, realistic constraints, engineering standards, and each group member's responsibility.

1.1 Motivation

There are a lot of auction websites that provide paying in the traditional ways, such as Visa and PayPal. After reviewing research and studying existing papers, we found the need for authentication process, scalability, and features such as a tracking system and improve user experience. In preparing our decentralized auction bidding system, we aim to address these challenges, making a decentralized auction bidding system more transparent and efficient. We believe that the advantages of blockchain will help to solve this problem.

1.2 Project Overview and Objectives

In this project, we aim to design and implement a Decentralized Auction Bidding System based on the blockchain that will prioritize a user interface with a focus on auctions. It will support transparent and highly competitive sealed auctions for single-item. Sellers will be able to effectively showcase their goods and services in forward auctions, creating competition among a broad audience of potential buyers. Additionally, the platform embraces single-sided auctions for ease of participation and understanding, allowing users to browse, and bid without the complexity of managing interactions between buyers and sellers. Our system stores user credentials and the user location. Operating as an Incentive Lottery, our website encourages customer engagement in the bidding process by offering the chance to win prizes through a lottery, serving as a promotional and marketing tool to attract and reward users.

1.3 Design Requirements

The following are the design requirements that will be implemented in this work:

1- The system shall be implemented using Ethereum smart contract with Base Sepolia blockchain and Bitcoin.
2- The system shall be implemented with a user authentication system to ensure that only registered users can place bids.
3- The system shall allow integrate reliable oracles to fetch real-time data for events and match outcomes, ensure oracles are secure and tamper-proof to maintain the integrity of the bidding process.
4- The system shall provide a friendly user interface displaying Ethereum wallet integration and bidding options and provide clear instructions for users on how to interact with Ethereum smart contracts.
5- The system shall provide bidders with automatic notifications, including shipping updates and tracking numbers. Shipping status shall be visible on the user dashboard.
6- The system shall inject an element of excitement that shall let bidders automatically participate in a lottery system using Robust randomization or similar algorithms.

1.4 Realistic Constraints

1. Economic constraints.

Our work will only use open-source solutions offering valuable, accessible sources such as cost-effective development, no licensing fees, extended lifespan, and resource efficiency as Solidity, MetaMask, MongoDB, React, and Node JS. Which anyone can use with just having access to an internet connection through a browser.

2. Manufacturability and Sustainability.

Scalability could be achieved and maintained through different blockchain mechanisms such as sharding, consensus mechanism selection, batching transactions, and Off-chain bidding (these terms will be discussed in detail in the following chapter).

1.5 Engineering Standards.

ISO/TC 307 (International Organization for Standardization Technical Committee 307: Blockchain and Electronic Distributed Ledger Technologies) [19]: Incorporating ISO/TC 307 standards into a decentralized auction system provides a foundation for improved security, interoperability, and compliance. This association helps create a more efficient and trustworthy environment for participants, encouraging the growth and adoption of decentralized auction platforms globally; based on our knowledge. Still, Ethereum is not considered by the ISO standard.

ERC-20 TOKEN STANDARD: Ethereum is considered under the ERC (Ethereum Request for Comment) standards, which are a series of standards. Among these standards is the ERC-20, which has a property that makes each Token the same (in type and value) as another Token, meaning that one Token is and will always be equivalent to all the other Tokens. In other words, it is a standard for Fungible Tokens [21].

ISO/IEC 27000 family (Information Security Management) [20]: provides a framework for establishing, executing, supporting, and continually improving an information security management system (ISMS). Incorporating the ISO/IEC 27000 series into a decentralized auction system establishes a strong foundation for information security, establishes user trust, mitigates risks, and demonstrates a commitment to the highest security management standards. This approach contributes to the overall resilience and integrity of the decentralized auction platform.

1.6 Preliminary Design

The initial design is represented in this section. As shown in Figure 1, the primary design components of this project are Sign-in, Auction, Lottery, Smart contract, and MetaMask. Sign-in authenticates bidders by obtaining their credentials and saving them in MongoDB. MetaMask is required for bidders to be able to bid. A smart contract is the core of our system, managing bids, auction rules, and transactions. The Auction section explains our system's auction rules and bidding process, while the Lottery section describes the lottery process and how the winner is selected. All these 5 components form a decentralized auction system. They are discussed in detail in Chapter 3.

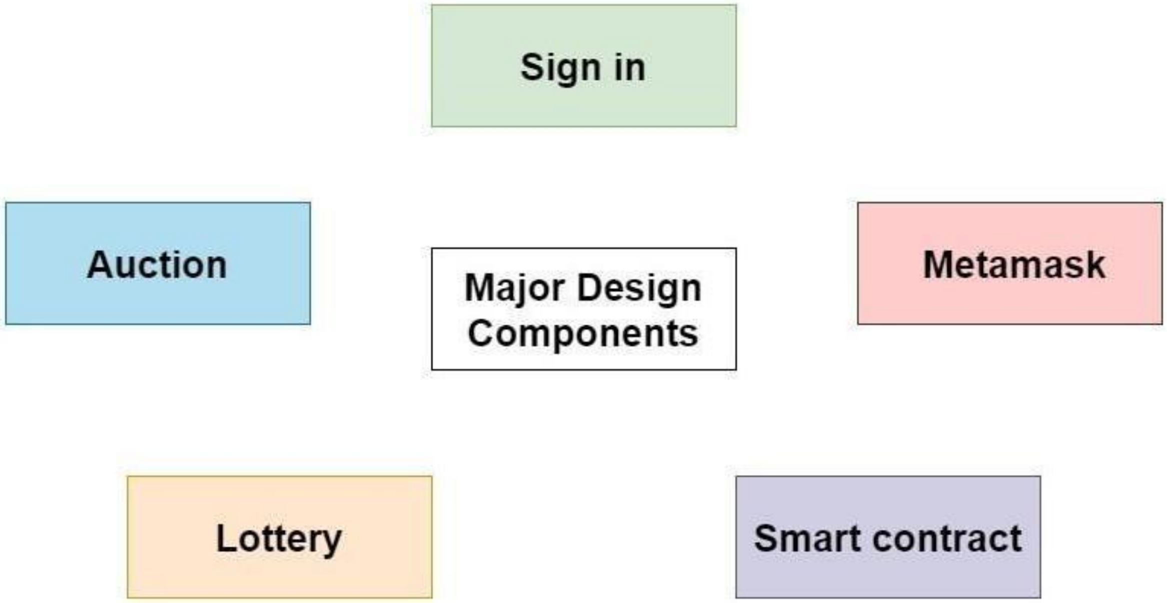


Figure 1: The major design component of the blockchain biding system.

1.7 Individual Responsibilities

Table 1: Group responsibilities.

Tasks	Omar	Ahmad	Mohammed
Documentation	✓	✓	✓
Literature Review	✓	✓	✓
Preliminary Design	✓	✓	✓
Design	✓	✓	✓
Implementation	✓	✓	✓

1.8 Report Organization

The rest of this report is organized as follows. The background in section 2 will introduce the main terms that will be used in our system model; furthermore, the literature review will cover the previous related works and their challenges. Section 3 includes design, section 4 evaluation of the system. Finally, section 5 the conclusion and future work.

2 Background & Literature Review

This chapter discusses two sections: background and literature review. The background contains subsections that introduce the definition of auctions, lotteries, blockchain, smart contracts, Oracles, and authentication. The literature review contains recent findings in the decentralized auction bidding systems field.

2.1 Background

Our decentralized system involves an auction module that introduces different bidding process roles of buyers and sellers. Simple and complex lotteries are integrated to immerse users in the bidding process. The backbone of our project is blockchain technology, mainly Ethereum Smart Contracts, providing security and automation in online auctions. Trusted oracles, like Chain-link, bridge blockchain with real-world data, ensuring accurate outcomes. User experience and security are prioritized through various authentication techniques, including password-based, multi-factor, biometric, token-based, and social login (OAuth). These elements will be discussed in the next part.

2.1.1 Auction

One approach to show products or services through competitive bidding is via auctions, which can be classified based on the bidding process, the roles of sellers and buyers, and the number of participants involved [1].

Bidding Process:

Open-Outcry Auction vs. Sealed-Bid Auction: In an open-outcry auction, bids are visible to all participants, encouraging increased competition. Conversely, sealed-bid auctions involve secret submissions, with bids known only to the auctioneer until the auction ends [1].

Roles of Buyers/Sellers:

Forward Auction vs. Reverse Auction: In a forward auction, sellers offer goods or services to multiple possible buyers. Meanwhile, a reverse auction flips the roles, with buyers competing to purchase from various sellers [1].

Participants:

Single-Sided Auction vs. Double Auction: A single-sided auction involves one group of participants (buyers or sellers) interacting with the auctioneer [2], who centrally determines prices. Meanwhile, a double auction engages buyers and sellers actively, with prices determined through the interaction of these two groups without centralized authority setting prices.

2.1.2 Lottery

Lottery models come in two main kinds: simple and complex [3]. Simple lotteries involve participants paying to join and winning prizes. Complex lotteries also require fees, but the prize distribution consists of chance-based steps.

There are different subtypes of lotteries for various purposes. Private Society Lotteries are for club members, Work Lotteries are for employees, and Residents' Lotteries are for people in specific areas. Incidental Lotteries are part of events. Incentive Lotteries use prizes to motivate engagement. Small Society Lotteries are for nonprofits with particular purposes, while Large Society Lotteries are for bigger organizations with more regulations. Local Authority Lotteries, run by governments, support local causes [3].

2.1.3 Blockchain

Blockchain, a fundamental concept in digital ledger technologies, is necessary in modern decentralized systems. At its core, a blockchain is a distributed digital ledger that records transactions and data in a transparent, secure, and decentralized manner [4]. Blockchain operates on a computer network, often called nodes, which collectively validate and store new transactions. These transactions are grouped into blocks and then sequentially added to the existing chain of blocks, forming a chronological and immutable record of all activities on the network. At its core, blockchain uses a consensus algorithm to show how the nodes can agree on the validity of the transaction. Different consensus algorithms serve various purposes depending on the blockchain network's goals and requirements, such as Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA) [1]. Each one of the consensus algorithms serves a distinct purpose. PoW, known for its decentralized nature, involves miners solving a complex mathematical operation for transaction verification. POS selects validators based on their cryptocurrency holding. In POA, validators are chosen based on their identity and reputation. PoW will be used for this project due to i ts decentralized nature, immutability, and security.

One of the blockchain features that could add a layer of scalability to the blockchain network is sharding. Sharding divides the network into small parts called shards. This means that not every node in the network is responsible for processing every transaction; instead, different nodes are responsible for other portions [15].

A distinct feature of blockchain is its security and immutability [1]. The data within each block is cryptographically tied to the previous one, making it highly challenging for any entity to alter historical transactions without the network's consensus. This trustworthiness extends to many applications beyond cryptocurrencies, including supply chain management, voting systems, bidding systems, and smart contracts, where the blockchain's capability to provide transparency and tamper resistance becomes necessary. In essence, blockchain technology offers a decentralized and secure solution for recording, verifying, and maintaining data across various industries and use cases, fundamentally changing how we conduct transactions and interact in the digital age [4].

2.1.4 Ethereum Smart Contract

Ethereum is a blockchain-based platform that authorizes the creation and implementation of smart contracts and decentralized applications [2]. This feature makes it a resilient and secure choice for online auction platforms, particularly in managing bidding and payment transactions. Smart contracts on the Ethereum network operate as self-executing software programs that execute the terms and conditions of auction agreements between buyers and sellers. These contracts are securely stored on the blockchain and can fully automate the auction process, from bid verification to the secure release of funds and the transfer of ownership for the sold item.

The smart contract can be crafted using different programming languages like C++, Ruby, and Solidity. [14] in this report, we use Solidity as a programming language for the smart contract within the framework of Remix's open-source tool [2]. This is because Remix IDE provides comprehensive tools for deploying, debugging, and testing Ethereum contracts. This specific contract was tailored for open-forward auctions, where the leading bid price is continuously disclosed during the bidding process. Its key components encompass the buyer's Ethereum Address (EA), the auction duration, the current leading bidder's address, and their corresponding leading bid amount. Importantly, bidders participating in this process maintain anonymity to each other.

Solidity is the language for building these smart contracts on the Ethereum blockchain. Smart contracts developed using Solidity define auction constraints, including different aspects such as bidding, payment processing, and the transfer of items. Their deployment ensures transparency [4], fairness and significantly reduces the risk of fraudulent activities and manipulation, rendering them highly suitable for online auction platforms.

2.1.5 MetaMask Wallet

Metamask wallet is an electronic browser extension wallet that is used for securing cryptocurrencies and tokens such as Ethereum (ETH), Tether (USDT), USD Coin (USDC), and various others [6]. It is known for its security features when users create their account on Metamask, Metamask provides 12-word secret phrase. These 12 words are essential for account recovery. If lost, access to the account might permanently be lost. It serves as a security measure to protect users' cryptocurrency assets and ensure secure access to their accounts [6].

2.1.6 Oracles

A trusted oracle network is crucial in decentralized systems since smart contracts cannot independently access external data. An oracle extracts real-time data and inputs it to the blockchain [2]. It acts as a bridge between blockchain technology and the outside world, encompassing payment systems, Internet of Things (IoT), Application Programming Interfaces (APIs), and other blockchains [2]. This functionality significantly broadens the scope of smart contracts, especially i n applications like auctions. Using multiple oracles within a network enhances the reliability of outcomes, ensuring that the data supplied to the contract remains accurate and that blockchain participants can trust in the correct execution of the contract [2].

An oracle network can be one centralized unit or a decentralized network of oracles. In a centralized oracle network, a single oracle or a controlled group is responsible for supplying external data to the smart contract, which has disadvantages such as lack of trust, security risks, and potential for manipulation. This approach involves multiple independent oracles operating together to extract real-time data from the smart contract. The primary advantage of a decentralized oracles network is that it offers resilience and redundancy; the responsibility is shared across multiple nodes. If failure happens or provides inaccurate data, the overall integrity of the Oracle network can be maintained through the consensus of the honest nodes. This is why, this project will focus on using the decentralized oracles network, particularly regarding integrity, reliability, and resilience.

This reliability is achieved by enabling each oracle within the network to support the same type of request, thus eliminating reliance on a single node for execution. The real-world data retrieval process adheres to a protocol that incentivizes truthful reporting by nodes and penalizes those that provide false information. As shown in Figure 2, a decentralized Oracle network, such as Chain-link, is a website that retrieves trusted real-time data into the auction system and offers a system overview of a forward auction process using a smart contract. [2].

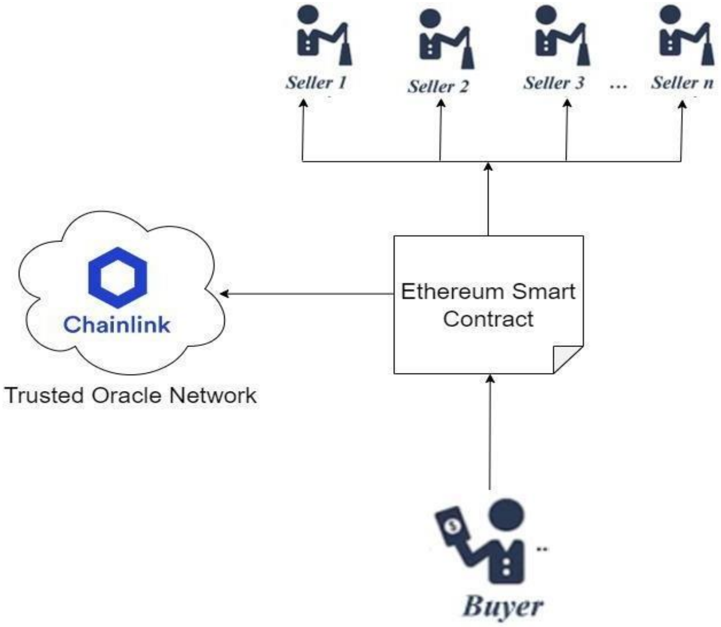


Figure 2: Forward auction using smart contracts and chain-link oracles.

2.1.7 Authentication

There are many ways to authenticate a client, such as password-based, multi-factor, biometric authentication, Token-based authentication, and Social Login, which is discussed in this part.

Password-based authentication: The simplest type of authentication requires entering a username and password to access an account. The least secure authentication is password-based despite being comparatively simple to set up.

Multi-factor authentication (MFA): MFA requires users to give two or more factors to authenticate their identity, adding additional protection to authentication. Passwords, one -time passwords (OTPs), and biometric elements like fingerprints are a common MFA factor. However, MFA can be more challenging to set up and use than password-based authentication.

Biometric authentication: Biometric authentication verifies user identities using distinctive physical traits, facial features, and fingerprints. Although biometric authentication is safer than other forms, it can be more costly and challenging to establish.

Token-based authentication: This approach uses hardware or software tokens to generate one-time passwords (OTPs). When a person logs in to an account, OTPs are used to confirm their identity. However, token-based authentication can be more expensive and complex.

Social Login (OAuth): Users can access a service without creating a new account using pre-existing social network accounts, such as Google or Facebook.

In this project, we implemented a single-sided sealed forward auction model integrated with an incentive lottery prize for the participants of the bidding process. Moreover, User authentication involves a username and password. Furthermore, the Oracle network is used to extract real-time data.

2.2 Literature Review

Many works cover a decentralized auction bidding system. In this section, we have considered some recent results in this field.

The work in [1] shows a complete and up-to-date analysis covering two areas of this study: the combination of blockchain technology with auction models. They considered the existing solutions for both fields, outlining their strengths and drawbacks. They also offer specialized classifications to facilitate better understanding. The advantages of integrating blockchain include increased security, trust-building elements, and the potential for decentralization. However, these merits are balanced by different challenges related to scalability due to the decentralized nature, consensus algorithms, and block size limitations due to technical complexities for solving complex mathematical operation s to mine blocks. Contrariwise, auction models perform efficiently in determining prices and promoting competitive bidding but suffer from problems related to uneven information. The potential for this is the fusion of blockchain and auction models brings about heightened trust and the creation of unchangeable records but requires addressing technical complexities and scalability issues. This survey aims to guide future research efforts in integrated blockchain-auction models to fully realize their potential by showing the various advantages and challenges within these domains.

In [5], an innovative application aims to overcome three significant challenges. As the company owns the infrastructure and charges fees for listing products and commissioning sellers, sellers are at the company's mercy. Profit margins are reduced, or buyer prices are increased as a result. There is no ownership of data by platform users. These companies own all reviews, purchase history, and personal information. By leveraging the Ethereum blockchain platform is built. They have created this application using the Truffle framework. This framework allows the development of decentralized applications (dapps) on the Ethereum blockchain, employing an Ethereum smart contract to simplify processes and reduce the need for intermediaries. User inputs are collected via a user-friendly web interface and sent to the Ethereum network using the web3.js API. Their assessment of the Rinkeby test network reveals that the application boasts an average transaction runtime of 3.8 seconds. However, the work was not compared to any other study, highlighting the trust and transparency benefits of Ethereum. However, it is worth noting that the application does exhibit a relatively high average gas consumption of 4.6 wei, which could raise cost concerns for users, particularly in resource-intensive applications. While our findings from the Rinkeby test network are valuable, it is essential to conduct further evaluation for real -time or high-frequency applications in a real-world context.

The paper [2] introduces an extensive framework for decentralized auctions, using Ethereum smart contracts to ensure transparent bidding, decentralized storage for securing bid-related documents, and trusted timer oracles for accurate timekeeping. The proposed solution of work in [2] is supported by specific steps buyers provide auction details, sellers submit necessary documents for participation, buyers' evaluation process, qualified sellers bid during the auction period, and the winning bidder announces that governs the role of smart contracts in auctions. The merits of this approach encompass enhanced transparency, secure document management, and reliable timing, all contributing to a more reasonable and efficient auction environment. Nevertheless, it is crucial to acknowledge potential challenges, including the complexities of implementation, reliance on external oracles, and cost considerations associated with deploying and executing smart contracts, which necessitate careful oversight and financial planning for participants in auctions.

A forward auction website has been created that utilizes Ethereum as its currency in paper [6]. This platform is built upon the Ethereum blockchain, employing Solidity to implement smart contra cts and integrate with MetaMask as a wallet browser extension. However, the system has certain drawbacks, specifically a complex user interface and a requirement for bidders to increment their bids by a predetermined amount. This incremental bidding approach can pose challenges for participants in securing the most favorable price for the auctioned item.

As mentioned in [7], a proposed Agri-auction system aims to bring farmers into an open market environment, allowing them to bid on any agricultural product in any quantity. The platform ensures anonymity by revealing only the highest bidding value and shows the relationship between gas fees and the number of bidders. The system also involves remote monitoring, controlling, and preventing catastrophes in agricultural settings. However, a considerable limitation is that the system still needs to be implemented, lacking a website and a mobile application. Additionally, it exclusively focuses on agricultural products.

The creation of an online purchase system that integrates blockchain technology is described in [8]. This online purchase system uses tools like PHP, JavaScript, HTML, CSS, and jQuery for different system elements. With additional features like bidder and seller user profiles, the system successfully supports contract signing, item placement, and procurement execution. Advantages include improved personalization through user profiles, responsive design, enhanced security, and transparency through blockchain integration. However, certain limitations are mentioned, such as a focus on basic purchasing processes, dependency on Google Chrome, vulnerability to slow internet connections, and constraints on purchasing items. Recommendations for improvement involve enhancing user-friendliness cross-browser compatibility, automating Know Your Customer (KYC) verification, expanding variables for procurement winner selection, and distinguishing between individual users and private companies. In summary, this system offers functional solutions for purchasing needs but requires improvements to address limitations and improve the user experience.

The authors of [9] introduce a sealed-bid e-auction scheme employing blockchain, smart contracts, bulletproof zero-knowledge proof protocols, and the Pedersen commitment algorithm, addressing challenges in bidder privacy and transaction fairness. The system ensures result validity and secure blockchain recording, allowing independent bidder result verification through zero -knowledge proof protocols. The performance analysis reveals that execution time for open and finish operations grows significantly with an increasing number of bidders, prompting the need for algorithmic enhancements in these phases. However, for publish, bid, and verify procedures, the execution time remai ned stable with different numbers of bidders. The paper compares the proposed mechanism with existing blockchain-based schemes, such as those by Galal and Youssef [10], Peng et al. [11], Xiong [12], and Yu [13], evaluating decentralization, anonymity, authentication, unforgeability, nonrepudiation, and winning price validation. The comparison highlights the proposed scheme's strengths in achieving a fair, secure, and reliable sealed-bid auction without a third-party auctioneer, establishing a decentralized auction with the help of blockchain technology. The paper also indicates the need for algorithmic improvements in auction phases, testing in real-world environments, and exploring alternative blockchain technologies other than Ethereum.

The proposed application in the paper [17] uses blockchain technology to sell used cars through a decentralized bidding application. The paper addresses the challenges of trust issues in auction used car centralized systems, vulnerability to scams, lack of transparency in bidding processes, and security issues in online car applications. The application aims to create a secure bidding application using blockchain that supports buying and selling used cars. Also, the platform aims to have procedures that allow the posting of ads. The proposed application describes the roles of users and integration with technologies like Web3, MetaMask, and IPFS (distributed file system) that improve transaction costs. The paper highlights the testing phase for smart contracts and the trust features achieved. Future work involves optimizing smart contracts for reduced gas consumption through sidechains/rollups and improving transaction efficiency within Web 3.0.

The paper [18] introduces a blockchain-based solution for the e-bidding system to solve two main problems: first, aiming to remove the third-party intermediary role by applying blockchain technology to preserve integrity and confidentiality in the auction process, Secondly, bidders lack a method of to guarantee that the leading bidder will not disclose their bidding price before the official unveiling. However, a considerable limitation is that this system uses the consensus mechanism 'solo,' a single-node consensus mechanism provided by hyper ledger fabric unsuitable for a production system. Recommendations for improvement involve enhancing the consensus algorithm to 'Kafka' or 'Raft,' both based on byzantine fault tolerance.

The research in [19] introduces Cream, a decentralized collusion-resistant e-auction system using smart contracts on the blockchain. It aims to replace centralized auctioneers with a distributed system to ensure transparency, trust, and security among auction users. It addresses three main challenges: firstly, lack of trust among participants, which could be solved using blockchain to ensure transparency and trust among bidders; secondly, bidder collusion. Bidder collusion occurs when multiple bidders secretly cooperate to manipulate the auction results for mutual benefit. The research aims to design a smart contract that prevents collusion, random rounding, and profit estimation techniques. Thirdly, regarding the implementation of complexity and costs, the paper proposed a solution to optimize the auction algorithm for computational efficiency to minimize the costs of executing smart contracts. However, the principles of Cream have not yet been extended to other types of auctions, such as open auction.

The paper in [20] introduces a decentralized marketplace on the blockchain . It is designed to enhance customer bargaining and streamline e-procurement. It addresses three principal challenges: firstly, lack of customer bargaining power, which could be solved by aggregating customer proposals to increase order volumes; secondly, high transaction costs and time delays were solved by implementing smart contracts to automate transaction, payment, and order fulfillment, thirdly information asymmetry and lack of transparency was solved by using blockchain because of its immutability. The platform aims to enhance scalability by exploring ways to use layer2 solutions. Additionally, it aims to improve the integration by developing more seamless integration with external systems.

In the previous works, we found multiple challenges. These are:

- Weak user interface or no website implementation: This affects user experience as users may find it difficult to interact with the system.

- Minimal user authentication: Several studies only implement basic user authentication methods such as username and password but neither of them is hashed which leaves the security leaves of the system vulnerable.
- Lack scalability of the system: Increased transaction volume can overwhelm the blockchain which leads to slower transaction processing. Also, high demand leads to increased gas fees making it expensive to execute smart contracts.

In this project we are aiming to solve these challenges through:

- Executing a friendly user interface and complete auction website. React allows us to create an interactive UI that enhances user engagement. React provides us with features to create an interactive bidding and live feedback on user actions such as bidding updates, and countdown timer, these can be done by integrating interactive elements like bidding buttons, and items history display. This ensures an engaging bidding experience.
- Using a centralized database to save our users' credentials: Using a centralized database, in our case, is a much more effective approach. The disadvantages of using a decentralized database are significant for our security needs. This is because, in a decentralized database, user credentials would be stored on every node in the network, making them accessible to every node. This poses security risks. On the other hand, the user credentials would be stored in a decentralized database.
- Minimizing bidders and smart contract interactions to reduce gas fees and enhance scalability. This is achieved by managing the bidding process at the front end. Operations that can be executed on the front end will be handled there, while only essential functions, such as payments and operations that cannot be performed on the front end, are processed by the smart contract. This ensures better performance.

3 Design

This chapter represents the design with multiple subsections: design, design requirements, and constraints as well as their analysis, and the summary of different design approaches/choices to be used in the implementation.

3.1 Developed Design

As mentioned in our pre-liminary design, the major design components of this project are Sign-in, Auction, Lottery, Smart contract, and MetaMask. Figure 3 represents the big picture of the Sign-in and user authentication (green color), auction (blue color), lottery process (orange color), smart contract (purple color), and MetaMask (pink color). In the following part, a detailed walkthrough represents every subsection.

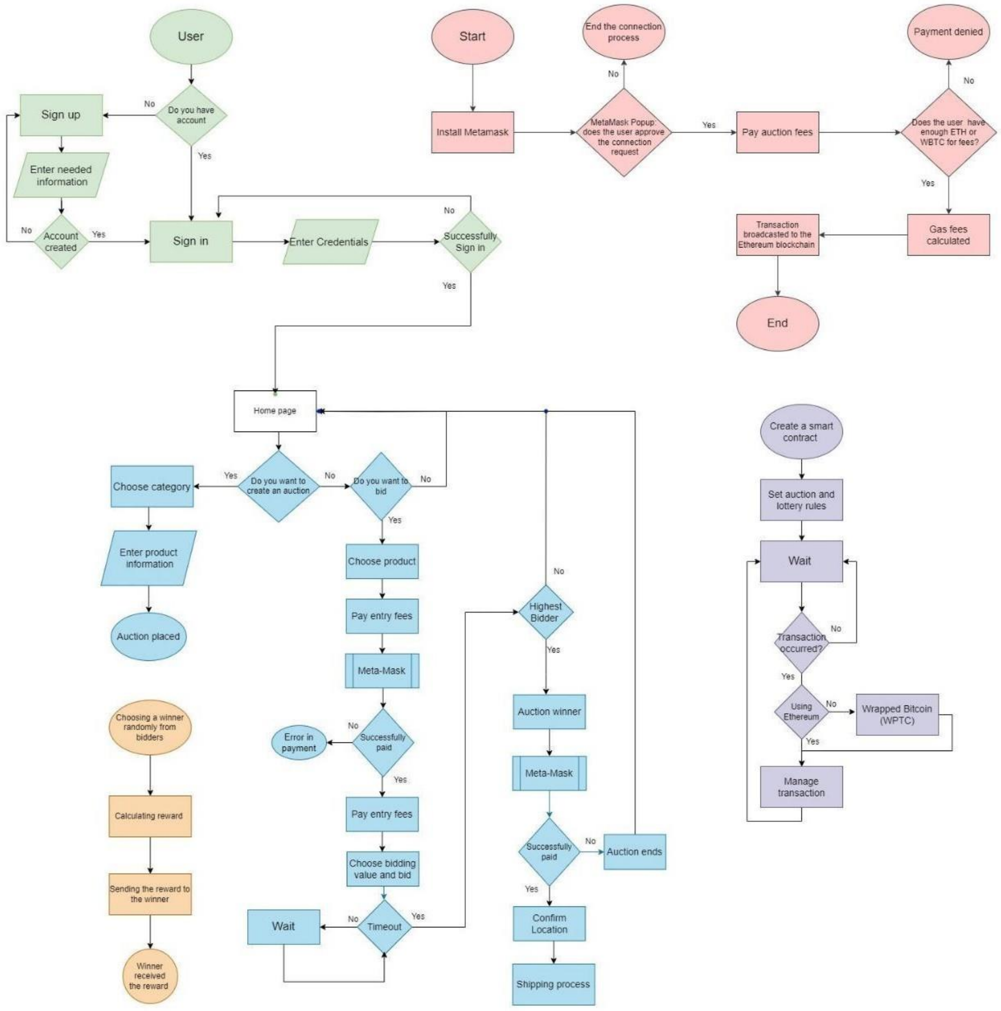


Figure 3: The design of the blockchain bidding system.

3.1.1 Sign-in and user-authentication

User authentication is an essential part of the proposed design. Figure 4 shows how users can create an account to access the website by entering the necessary information, such as username, password, and user location which will be done automatically by GPS. Users' password is hashed by the Bcrypt algorithm [25] and stored in the MongoDB database to maintain our users' credentials' integrity and confidentiality and to prevent unwanted access to their confidential information. If the user enters the correct password, the user is moved to the home page. Otherwise, if he entered an incorrect password, he would be asked to renter the password.

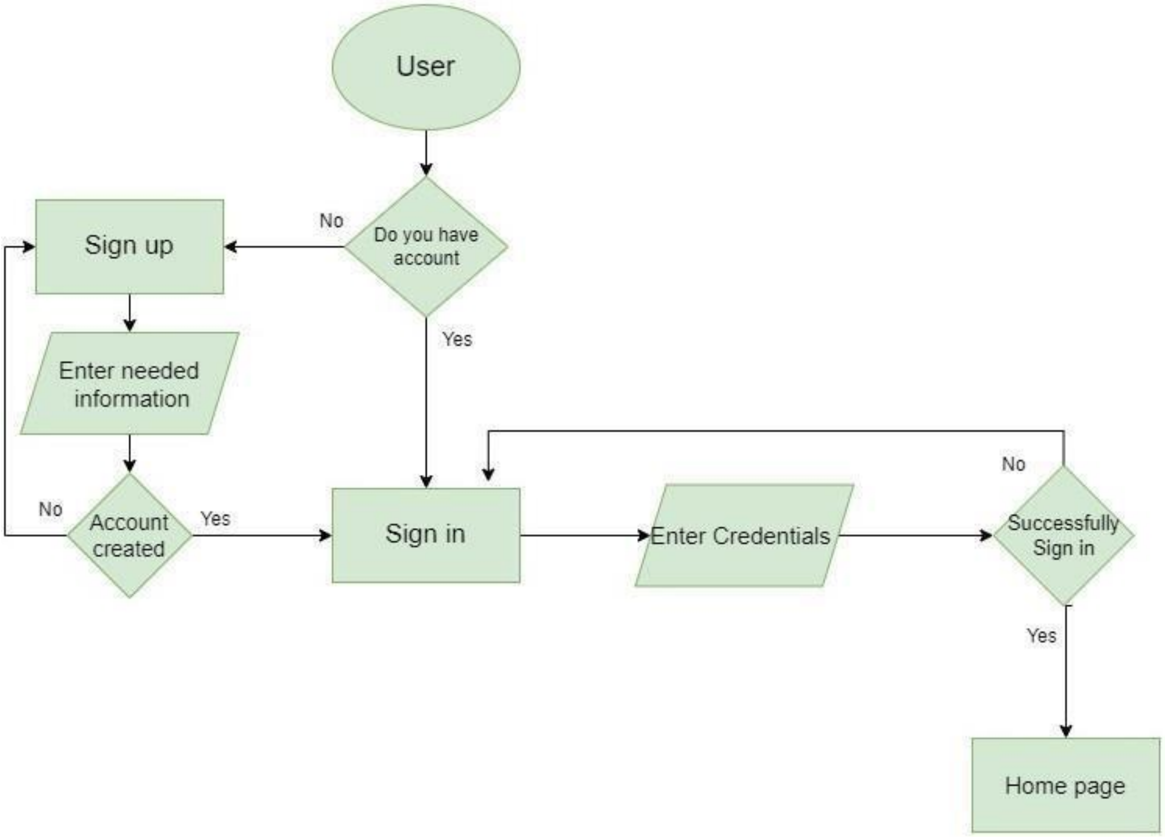


Figure 4: Authentication process.

3.1.2 Auction process

Figure 5 illustrates the auction process, which consists of two sides: sellers and buyers (bidders). On the seller side, the process starts by selecting a product category from a predefined category, product image, and the seller Metamask. Then, the auction is placed, and the seller can return to the home page. Alternatively, bidders can bid on a selected product after paying the entry fees through Meta-

Mask, if the payment is not completed, an error appears as a pop-up message. In addition, the bidder should bid the value he is willing to pay, and the bid value should be available in his MetaMask wallet. Through this sequential process, bidders should wait till the auction ends. After the termination of the auction, the bidder with the highest bid value wins by paying the bid value.

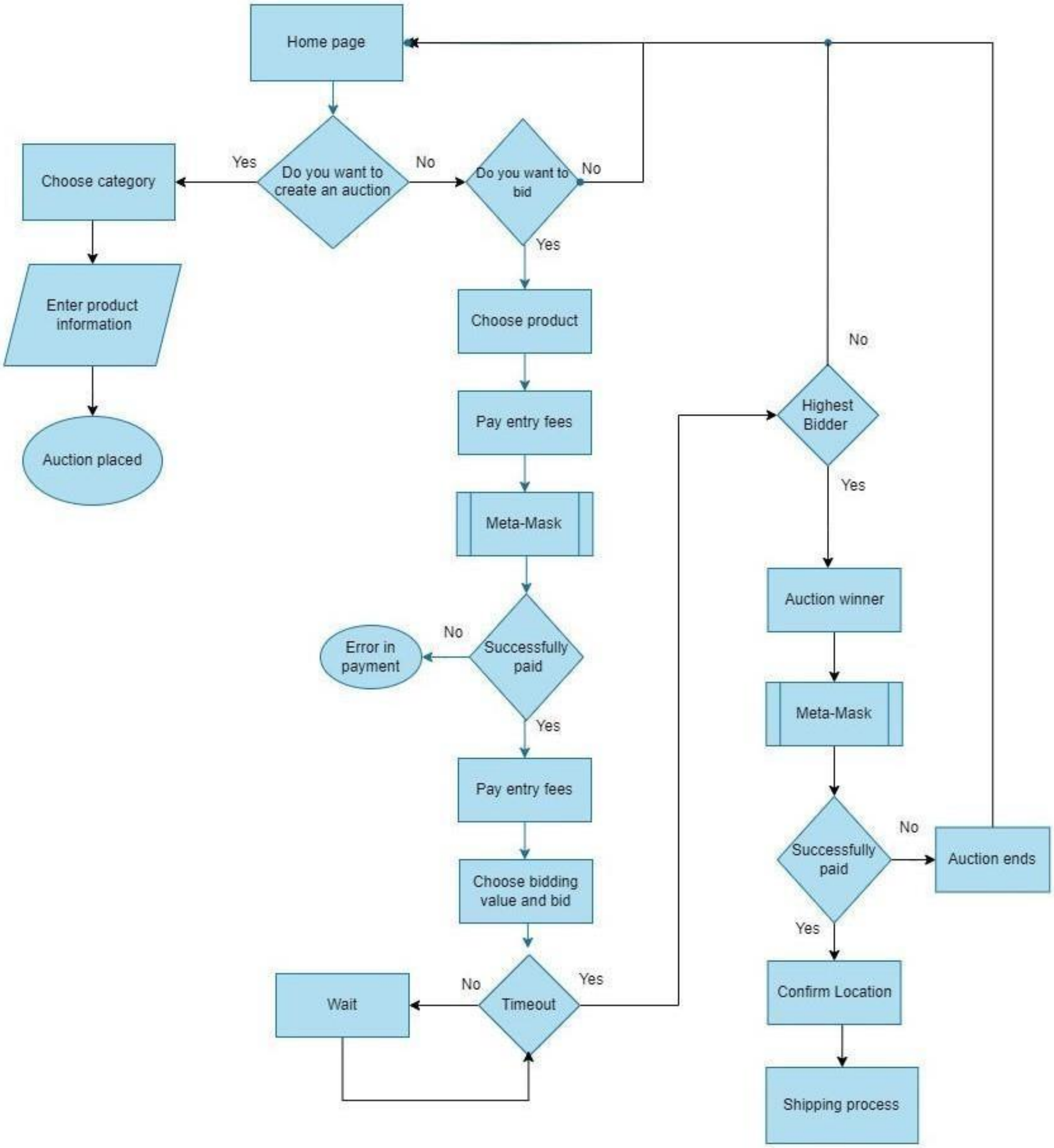


Figure 5: Auction process.

3.1.3 Lottery process

Figure 6 represents the lottery process. The lottery adds a spice of excitement to the auction process where each auction has its own lottery; auction participants automatically join the lottery by paying the entry fees. The winner is randomly selected, and the reward is calculated from a portion of entry auction fee summation.

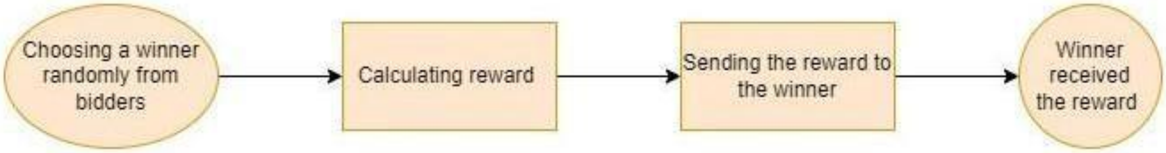

Choosing a winner randomly from bidders
Calculating reward
Sending the reward to the winner
Winner received the reward

Figure 6: Lottery process.

3.1.4 Smart contract

This part represents how the smart contract is created and how the transactions are dealt with through Figure 7. A smart contract is created by setting some rules for the auction and the lottery, waiting for a received or sent transaction to be managed, and retrieving the real-time price of Ethereum.

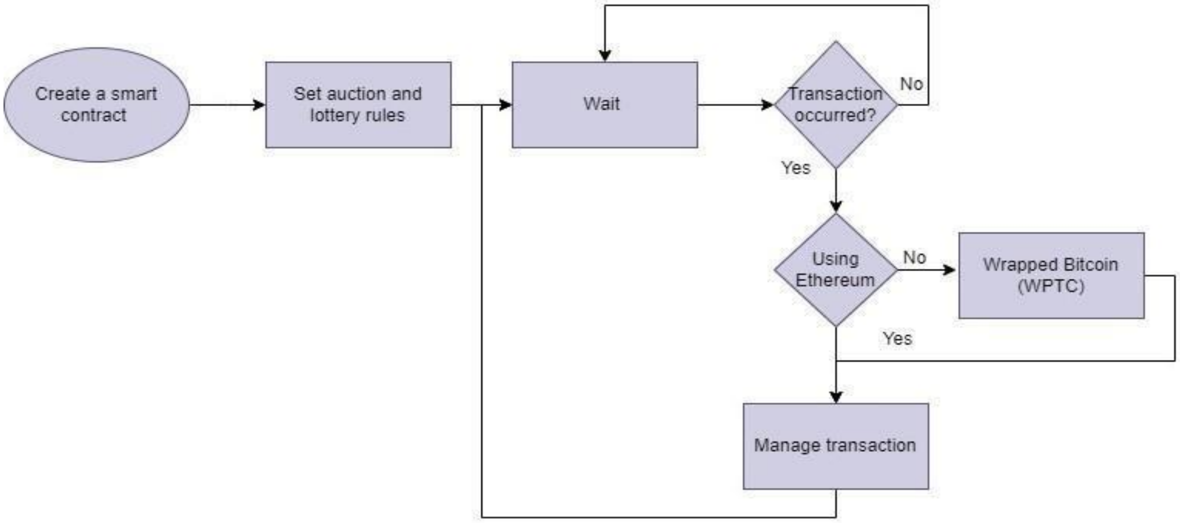


Figure 7: Smart contract.

3.1.5 Meta-Mask

In this part, the Meta-Mask process is discussed. Once MetaMask is installed, the User can gain access to the auction. As shown in Figure 8, the user is asked to accept the connection using a MetaMask popup window. The connection process ends if the User denies the request. Upon approval, the User must pay the auction fees, but first, the system must confirm there is enough Ethereum in their wallet. If there isn't enough Ethereum, access is rejected. Next, the system checks if the User has paid the auction fees. If payment is successful, the User can start bidding. After the auction terminates, the system confirms if the User is the winning bidder. If not, the User's payment is uncompleted. Gas fees are then computed for the transaction, and after the User signs the transaction using his private keys, it is broadcast to the Ethereum blockchain. Finally, bid confirmation is displayed.

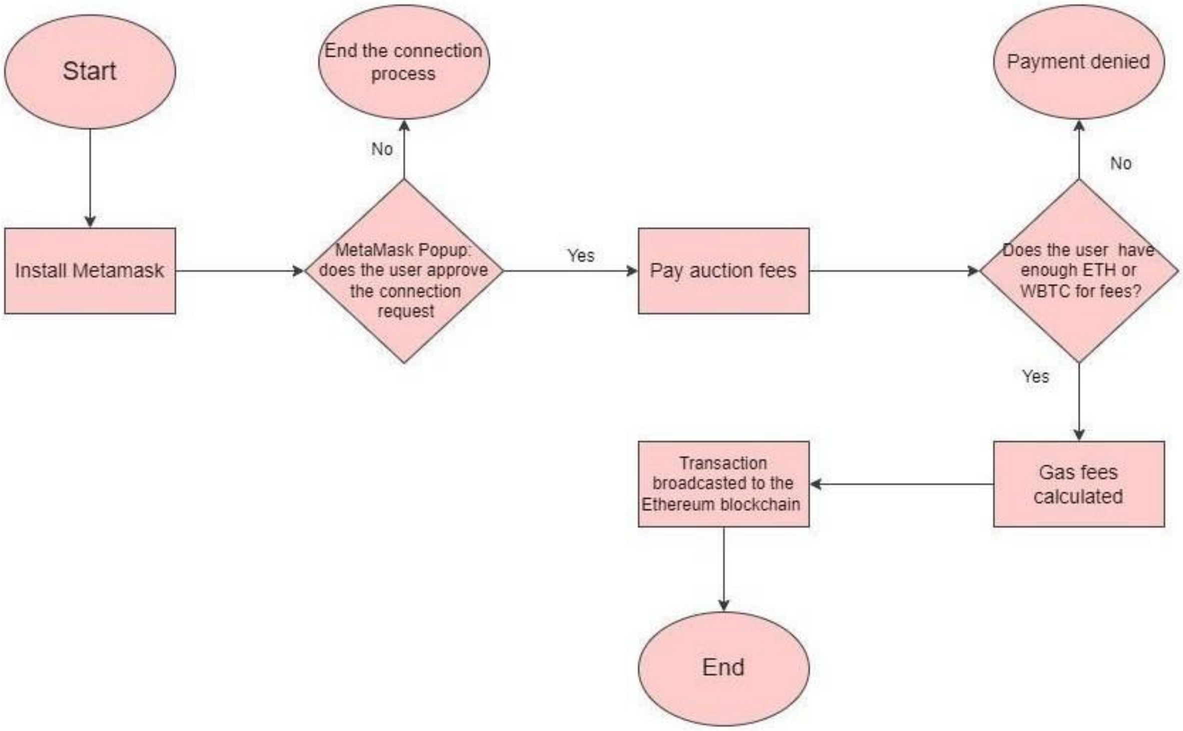

Start
Install Metamask
MetaMask Popup: does the user approve the connection request
No
End the connection process
Yes
Pay auction fees
Does the user have enough ETH or WBTC for fees?
No
Payment denied
Yes
Gas fees calculated
Transaction broadcasted to the Ethereum blockchain
End

Figure 8: MetaMask.

3.2 Design Requirements

As mentioned in Chapter 1, the design requirements are as follows:

1- The system shall be implemented using Ethereum smart contract with Ganache personal blockchain and Bitcoin.
2- The system shall be implemented with a user authentication system to ensure that only registered users can place bids.
3- The system shall allow integrate reliable oracles to fetch real-time data for events and match outcomes, ensure oracles are secure and tamper-proof to maintain the integrity of the bidding process.
4- The system shall provide a friendly user interface displaying Ethereum wallet integration and bidding options and provide clear instructions for users on how to interact with Ethereum smart contracts.
5- The system shall provide bidders with automatic notifications, including shipping updates and tracking numbers. Shipping status shall be visible on the user dashboard.
6- The system shall inject an element of excitement that shall let bidders automatically participate in a lottery system using Robust randomization or similar algorithms.

3.3 Analysis of Design Requirements

- **Base Sepolia personal blockchain (partially achieved)**: Our design implements an Ethereum smart contract with Base Sepolia personal blockchain. It will provide fake Ethereum for testing purposes.
- **User authentication (achieved)**: Our system implements a user authentication system to ensure that only registered users can place bids using username and password through MongoDB by storing the username and the hashed password. Then, the cookies will keep tracking the user through the bidding process.
- **Integrate reliable oracles (achieved)**: Our system integrates reliable oracles to fetch real-time data. This implies using chain links to get Ethereum's real-time price through the Ethers package on the front end.
- **Friendly user interface (achieved)**: We designed a friendly user interface through React to display bidding options and providing clear instructions for users on how to interact with Ethereum smart contracts through a straightforward user interface**.**
- **Shipping process (partially achieved):** Our system provides Google GPS service to store the user’s location. On the other hand, because lack of tracking test API we couldn’t do the full tracking system**.**
- **Lottery system (achieved):** The smart contract injects to the system an element of excitement that lets bidders automatically participate in a lottery system.

3.4 Analysis of Design Constraints

As described in Chapter 1, the design constraints are as follows:

1. Economic constraints.

Our work only uses open-source solutions offering valuable, accessible sources such as cost-effective development, no licensing fees, extended lifespan, and resource efficiency as Solidity, MetaMask, and Google GPS service. Which anyone can use by just having access to an internet connection through a browser. **This has been achieved through leveraging widely known technologies (Solidity, Metamask, and Google GPS) that empower users globally**.

2. Manufacturability and Sustainability.

Using React and Node.js for blockchain apps can improve manufacturability through its ecosystem. Also, these technologies support efficient performance and ease of maintenance. They offer solutions such as **sharding, consensus mechanism selection, batching transactions, and off -chain bidding.** These mechanisms provide efficient performance, reduced costs, and improved adaptability, which anyone can benefit from with an internet connection. By that, we can achieve high efficiency and sustainability throughout its lifecycle. We can maintain scalability and handle increased loads by leveraging cloud infrastructure. This method guarantees that the system remains cost-effective, resilient, and capable of supplying a constant user experience throughout its lifecycle.

3.5 Different Designs Approaches/Choices

In summary, our design approach is as follows: firstly, the smart contract uses the Solidity programming language and is compiled using Remix. We utilize the Base-Sepolia network to obtain fake Ethereum for testing our smart contract. The smart contract connects to the MetaMask wallet to facilitate payments. Additionally, we implemented the interface using React, HTML, CSS, and NodeJS. For shipping, we use Google GPS service to capture the winner's location.

Centralized vs decentralized database:

Centralized database offers better control over sensitive user credentials unlike decentralized databases, which send sensitive credentials all over the network. This increases security risk. On the other hand, centralized database offers efficient querying and retrieval of data, ensuring sensitive credentials are accessed quickly. This is why we have chosen the centralized approach.

Ethereum test networks and blockchain simulators:

Different environments exist, such as local blockchain simulators like Ganache, and public test networks like Base Sepolia and Goerli. However, the main issue isthat not all test networks support deploying the smart contract using Remix as the Ganache test network. Regarding Geroli, its support will be discontinued, so we have chosen the base Sepolia as our test network.

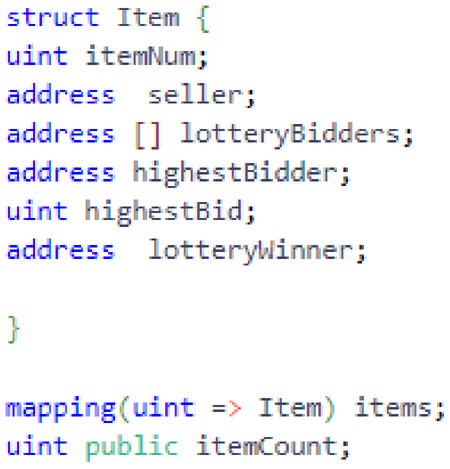
Remix, Hardhat, and Truffle.io:

Remix is an ideal web-based IDE tool for debugging and deploying blockchain. It also supports live contract interaction. In contrast, Hardhat and Truffle are more complex and frequently updated, making them harder to learn. Deploying a smart contract by only connecting via Remix/MetaMask does not use private keys stored in text files, ensuring a more secure deployment process. On the other hand, Hardhat and Truffle rely on local environment files that contain private keys for deploying smart contracts. Storing private keys in text files is a significant security risk, as it allows access to the wallet.

3.6 Implementation

In our Decentralized Auction Bidding System project, a smart contract is written in a solidity programming language, the Ethereum cryptocurrency program. Smart contract and front end manage the auction from A to Z, which chooses the winner among the participants and the lottery winner to get his prize. on the other hand, we used MongoDB as our centralized database management system that stores our participant's usernames and hashed passwords to ensure data protection, integrity verification, consistency, and performance [26]. MongoDB also stores detailed auction information. MongoDB is popular with Node.js, an asynchronous event-driven JavaScript runtime designed to build scalable network applications. Our frontend, framed using React, efficiently prompts the highest bidder to complete payment and display the auction to users. This design helps to minimize the cost of gas fees by reducing interactions between users and system to provide a great user experience.

3.6.1 Smart contract

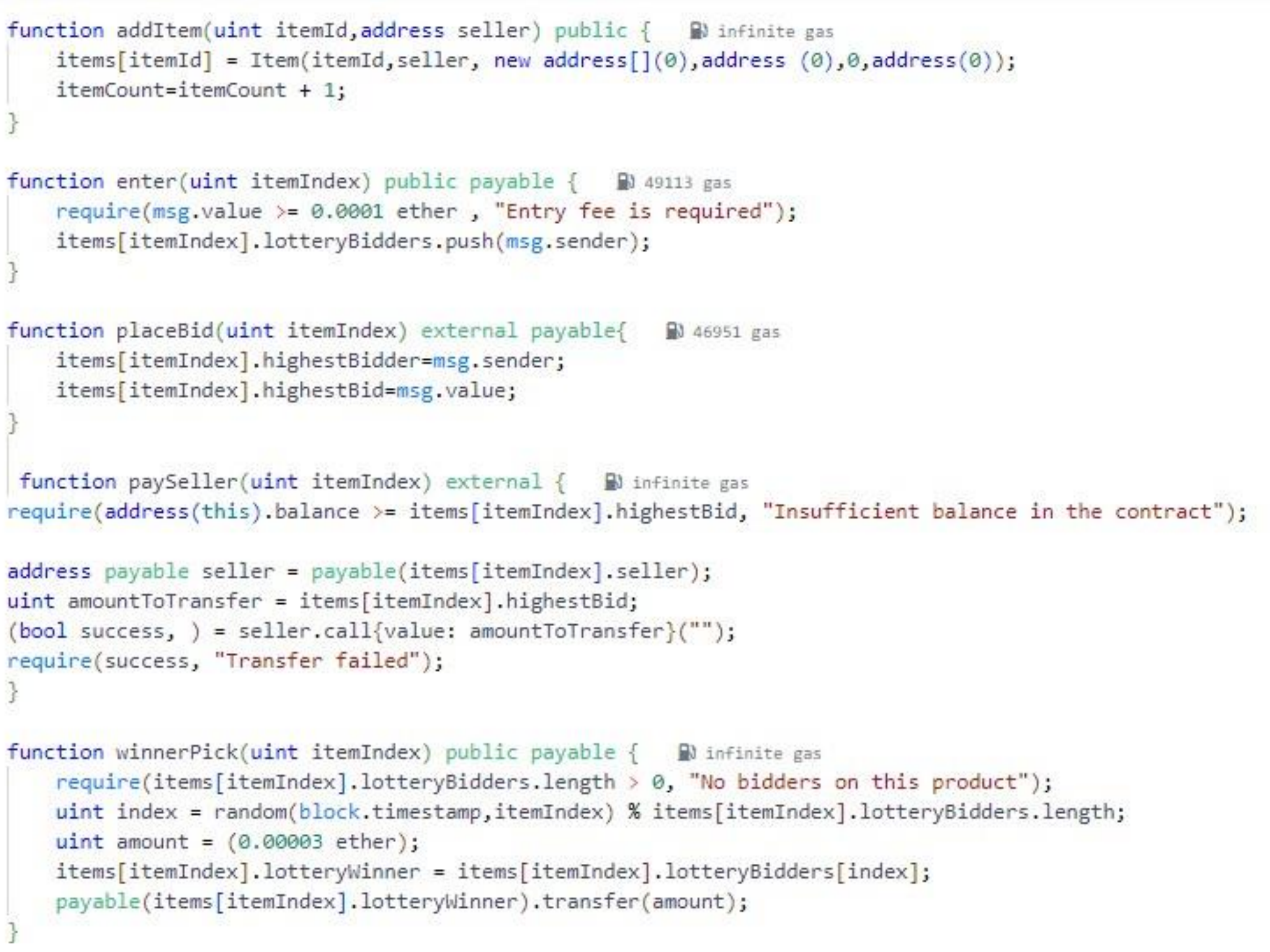


```
struct Item {
uint itemNum;
address  seller;
address [] lotteryBidders;
address highestBidder;
uint highestBid;
address  lotteryWinner;

}

mapping(uint => Item) items;
uint public itemCount;
```

Figure 9: Item's structure and some variables.

As shown in Figure 9, we started by building the structure of the items; each has a unique item number (ID), the seller's address, an array containing users who paid the entry fee, the address of the highest bidder, and the bid value. At the end of the auction process, a smart contract retrieves the address of the lottery winner and the item counter. Additionally, a mapping of items is created by connecting item numbers with their corresponding item, as using an array of items would require a fixed size.

Figure 10 shows the 'addItem' function, which takes two parameters: the item number and the seller's address; this function also increases the item counter. To enter the auction, the 'enter' function is implemented, which takes one parameter, the item number; this function allows users to join the auction and the lottery for the item by paying an entry fee of 0.0001 Ethereum.



```
function addItem(uint itemId,address seller) public {
    items[itemId] = Item(itemId,seller, new address[](0),address (0),0,address(0));
    itemCount=itemCount + 1;
}

function enter(uint itemIndex) public payable {
    require(msg.value >= 0.0001 ether , "Entry fee is required");
    items[itemIndex].lotteryBidders.push(msg.sender);
}

function placeBid(uint itemIndex) external payable{
    items[itemIndex].highestBidder=msg.sender;
    items[itemIndex].highestBid=msg.value;
}

 function paySeller(uint itemIndex) external {
require(address(this).balance >= items[itemIndex].highestBid, "Insufficient balance in the contract");

address payable seller = payable(items[itemIndex].seller);
uint amountToTransfer = items[itemIndex].highestBid;
(bool success, ) = seller.call{value: amountToTransfer}("");
require(success, "Transfer failed");
}

function winnerPick(uint itemIndex) public payable {
    require(items[itemIndex].lotteryBidders.length > 0, "No bidders on this product");
    uint index = random(block.timestamp,itemIndex) % items[itemIndex].lotteryBidders.length;
    uint amount = (0.00003 ether);
    items[itemIndex].lotteryWinner = items[itemIndex].lotteryBidders[index];
    payable(items[itemIndex].lotteryWinner).transfer(amount);
}
```

Figure 10: The main functions of the smart contract.

Next, the 'placeBid' function is created, which allows users who have paid the entry fees to place a bid by specifying the amount of Ethereum the user is willing to pay. This function, assisted by the front end, enables users to place bids for the selected item, helping to reduce gas fees. Additionally, the 'paySeller' function allows the smart contract to pay the seller the highest bid amount.

Moreover, the 'winnerPick' function randomly selects the lottery winner from an array of users, the winner receives a prize equal to three times the entry fee. Lastly, getters functions were implemented to return the necessary data for building the website.

3.6.2 Front-End Design Interaction

React is a web and native user interface library written in JavaScript [30] used to build our interface and interact with our smart contract. Ethers package is used in the front end to interact with the Ethereum blockchain [29]. Ethers allows us to extract the smart contract using smart contract API and the smart contract address and use the signer and provider during deployment.

3.6.3 Login and Sign Up

Figure 11 shows the signup process. We have two text fields that take the user's name and password and a signup button that injects the Google GPS into it to store the user's location, which will be used for the delivery process in the future. If the user entered a week password a pop up message will be received as shown in Figure 12. Furthermore, on the login page, we have two text fields for username and password as shown in Figure 13. If the user entered an incorrect username or password, an invalid username or password message will appear to the user as shown in Figure 14. otherwise, the user will land on the home page to choose between three different which will be discussed in later in these sections.

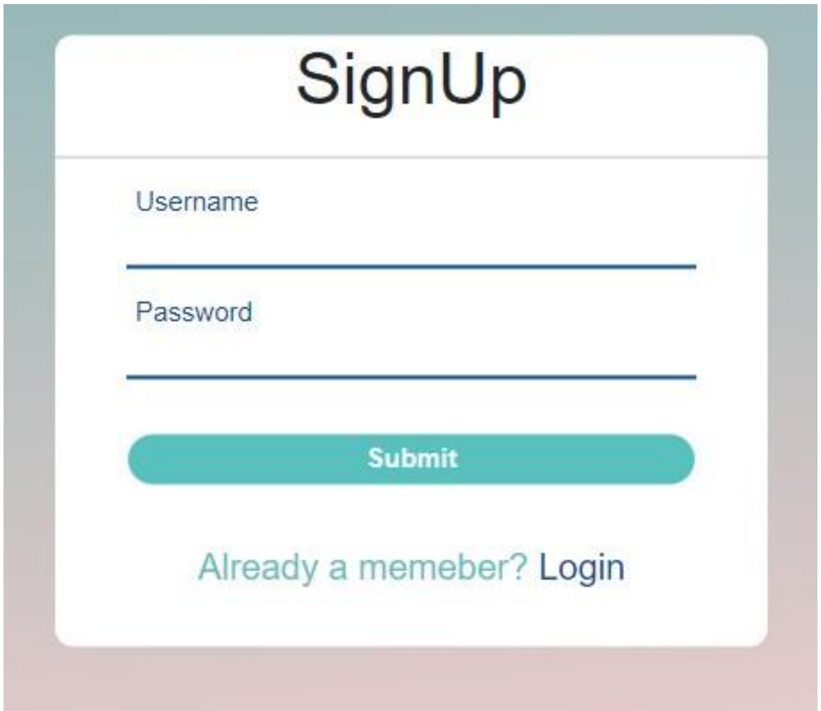
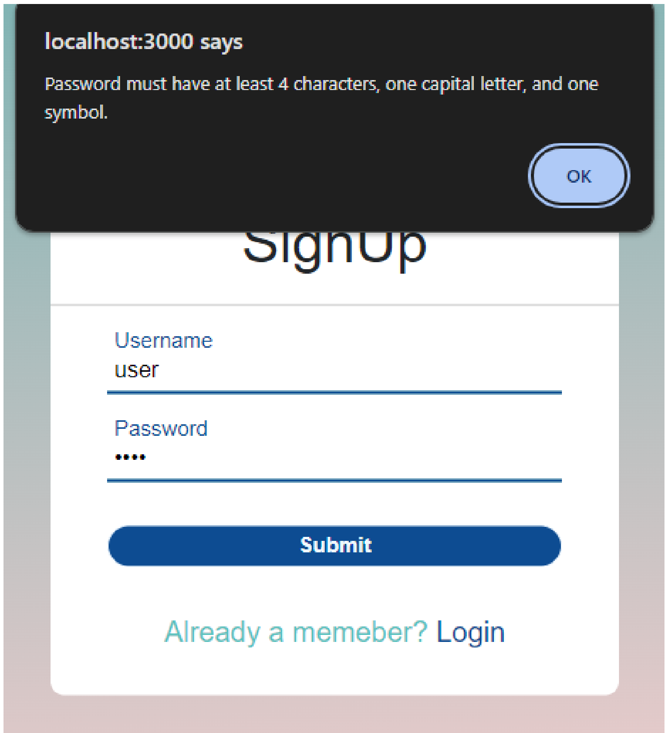


Figure 11: Sign-up page.



localhost:3000 says

Password must have at least 4 characters, one capital letter, and one symbol.

Username
user

Password
••••

Figure 12: Password validation alert.

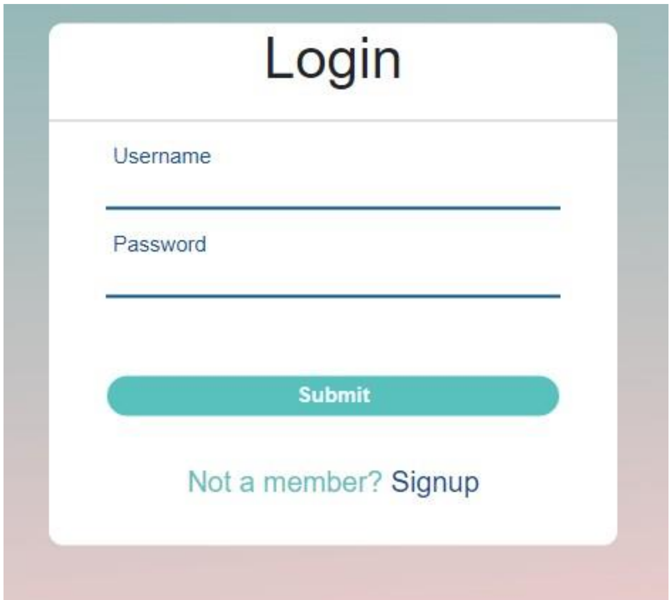
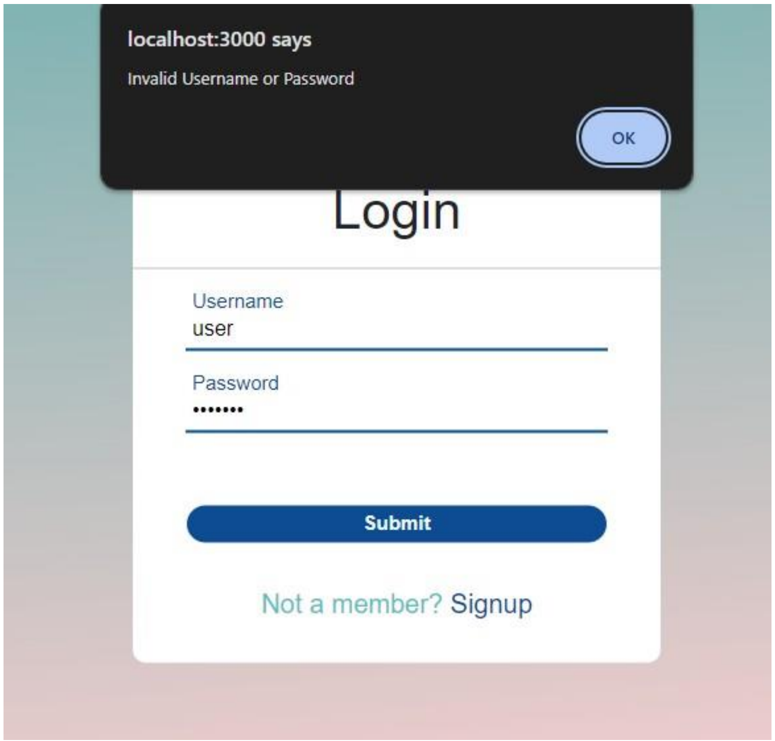


Figure 13: Login page.



localhost:3000 says

Invalid Username or Password

Login

Username
user

Password
•••••••

Figure 14: Wrong username or password alert.

3.6.4 Post Auction

As shown in Figure 15 the Post Auction page contains an upload button that will open a window to select the picture for the item. Also, it has a dropdown button that contains multiple category options such as phones, accessories, laptops, and others. In addition, an input section used to store numbers for the duration of time for the items. Finally, we have a text field to store the seller's MetaMask wallet address to pay the seller after the auction ends and a submit button that will upload the item to the database and will call the function add item in the smart contract using Ethers package to interact with the Ethereum blockchain [29]. Ethers allows us to extract the smart contract using smart contract API and the smart contract address and use the signer and provider during deployment.

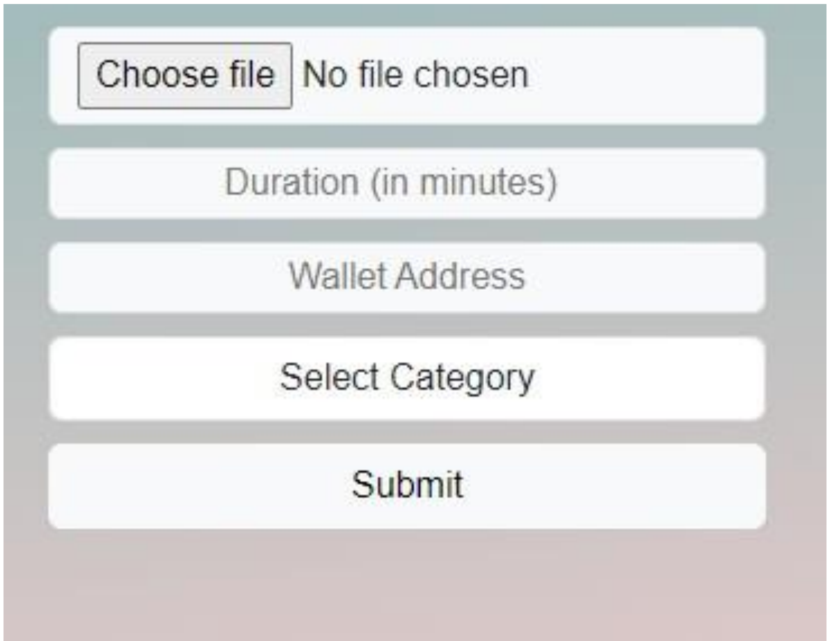
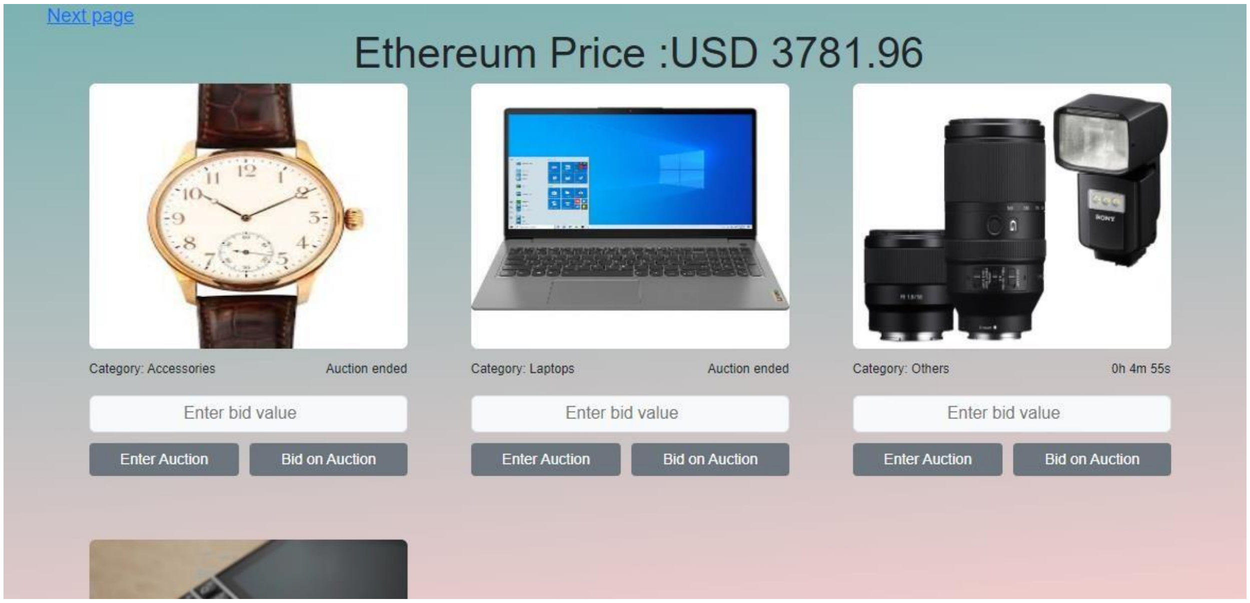


Figure 15: Post auction page.

3.6.5 Bid on Items

Figure 16 shows Ethereum's real-time price and a list showing the items with one text field and two buttons: enter auction and place a bid. The enter auction button will use Ethers to interact with the Ethereum blockchain [29]. On the other hand, the place bid button will take the number in the text field and manage bids without interacting with the smart contract to reduce gas fees.

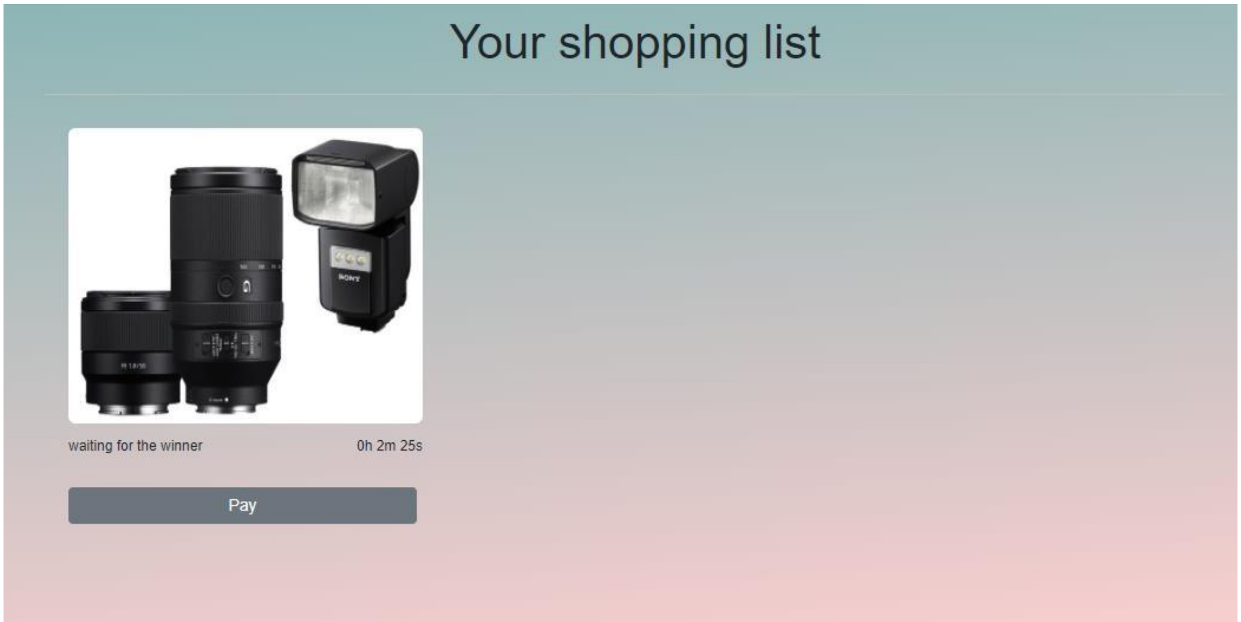


Ethereum Price :USD 3781.96

Figure 16: Bid on items page.

3.6.6 Shopping list

After the auction ends, we will have a page showing the entered auctions, allowing the user to pay the bid value entered in the bid on the items page using the pay button based on the user's auction outcome: winner or loser and the lottery results as shown in figure 17.



Your shopping list

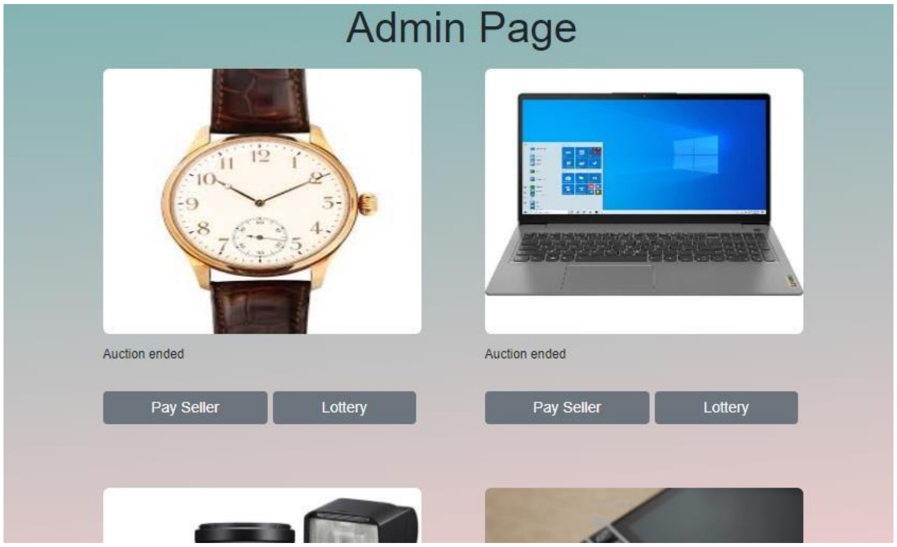
waiting for the winner

0h 2m 25s

Figure 17: Shopping list page.

3.6.7 Admin Page

Figure 18 shows the admin page, which can be accessed only by an admin user. It contains all items and functions only the admin can access, such as paying the seller and picking the lottery winner to complete the auction.



Admin Page

Auction ended

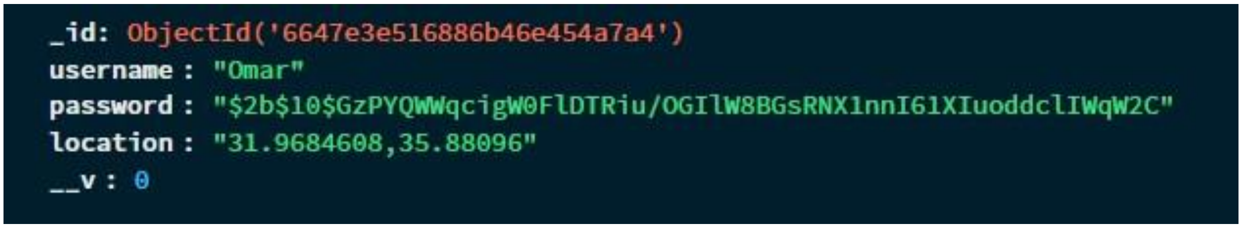
Auction ended

Figure 18: Admin page.

3.6.8 Database Design

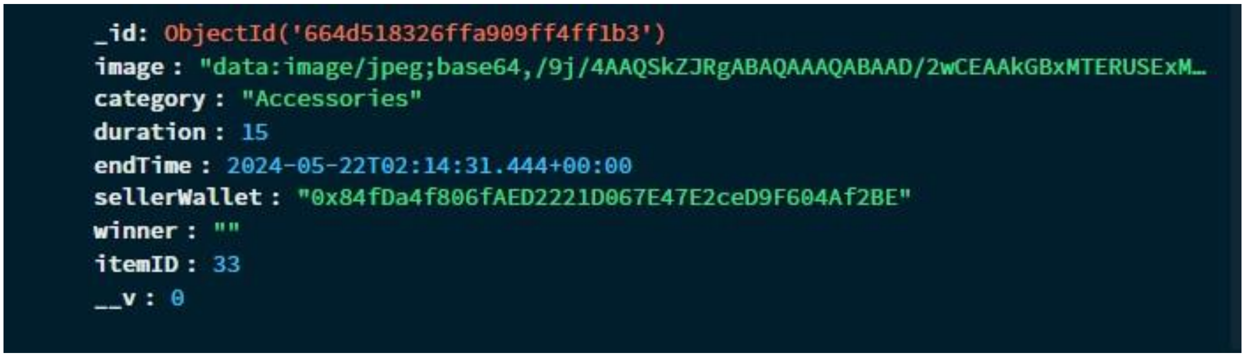
Our developed system relies on a NoSQL centralized database to store data using MongoDB through two collections. MongoDB was selected because of its flexibility, which aligns with our dynamic system.

The first collection is to store and manage user credentials, including usernames, hashed passwords, and locations all as strings as shown in figure 19. Figure 20 presents the second collection which handles images and auction info uploaded by sellers, which is dealt with by storing five fields. The image field stores a string representation of the uploaded image converted in the front end to base64. The category field specifies the category associated with the image and unique item numbers . We used two fields, duration, and end time, for the auction; the seller enters the duration variable, and then the end time field is calculated by adding the duration with the current time, and then it is stored.



```
_id: ObjectId('6647e3e516886b46e454a7a4')
username : "Omar"
password : "$2b$10$GzPYQWWqcigW0FlDTRiu/OGIlW8BGsRNX1nnI61XIuoddclIWqW2C"
location : "31.9684608,35.88096"
__v : 0
```

Figure 19: User credentials.



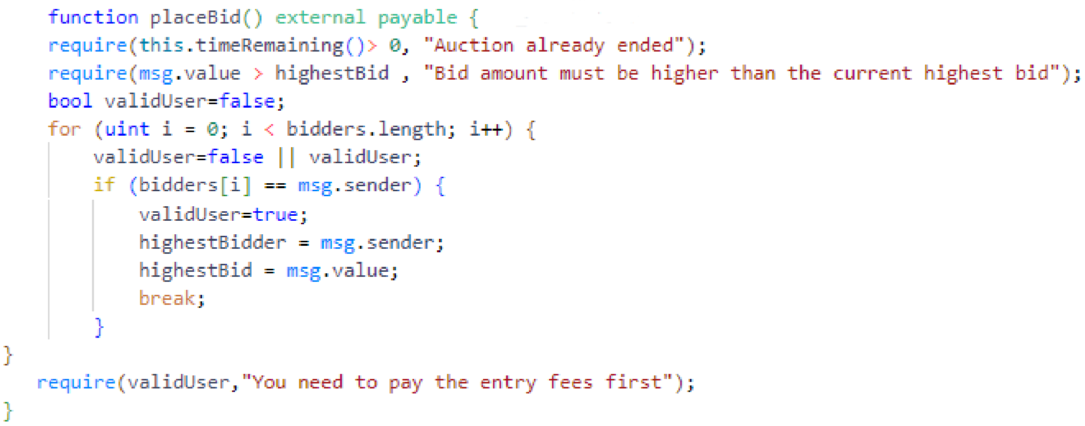
```
_id: ObjectId('664d518326ffa909ff4ff1b3')
image : "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBxMTERUSExM…
category : "Accessories"
duration : 15
endTime : 2024-05-22T02:14:31.444+00:00
sellerWallet : "0x84fDa4f806fAED2221D067E47E2ceD9F604Af2BE"
winner : ""
itemID : 33
__v : 0
```

Figure 20: Items collection.

4 Evaluation

This chapter measures the smart contract performance gas costs by executing the smart contract in a Base Sepolia test environment. As discussed in Chapter 3, our proposed approach manages bids for multiple items simultaneously. It is implemented by assigning a unique number to each item. As mentioned previously, our proposed approach consists with 5 methods ‘addItem’, ‘enter’, ‘placeBid’, ‘paySeller‘ and ‘winnerPick’.

To compare our work, we implemented another approach that depends on the smart contract fully to manage the bidding. This approach will serve as a benchmark to compare our proposed approach. Which consists of the same 5 methods but the ‘placeBid’ is the only method with a difference. In the benchmark approach ‘placeBid’ allows users to place bid and checks the new bid is higher than the current highest bid as shown in Figure 21. The main difference between the benchmark approach and our approach is that our approach takes leverage of the frontend to select the highest bid rather than implementing it in the smart contract as the benchmark approach. This results in reducing the code complexity, which eventually results in consuming fewer transaction costs and the increase of efficiency.



```
    function placeBid() external payable {
    require(this.timeRemaining()> 0, "Auction already ended");
    require(msg.value > highestBid , "Bid amount must be higher than the current highest bid");
    bool validUser=false;
    for (uint i = 0; i < bidders.length; i++) {
        validUser=false || validUser;
        if (bidders[i] == msg.sender) {
            validUser=true;
            highestBidder = msg.sender;
            highestBid = msg.value;
            break;
        }
}
   require(validUser,"You need to pay the entry fees first");
}
```

Figure 21: Shows the ‘placeBid’ method in benchmark approach.

Since our proposed approach uses the front end to manage the item highest bidder, it is expected that it will consume fewer transaction fees than the benchmark approach for several reasons. One of them is code complexity in which the benchmark approach has a more complex smart contract to handle the auction process than our proposed approach. Secondly, our proposed approach minimizes the interaction with smart contract that leads the reduction of the gas fees consumption.

To test our approach, we tested ten times at different times of the day and took the average. The results are summarized in Table 2. For the methods in the contract, the first method is "addItem" with an average gas cost of 45781, which is used to add an item to the website by the seller. The following method is "enter" with an average gas cost of 70240.4, which is responsible for adding users to the auction by receiving entry fees. Our third method is "placeBid", with an average gas cost of 29659.6, which is accountable for receiving bids from bidders. Our fourth method, "paySeller", with an average gas cost of 36917, is responsible for transacting the highest bid to the winner. The last method "winnersPick" method with an average gas cost of 40387.5, which is used to get the lottery prize winner.

Main Methods	Avg. Gas Transaction Cost	1st Run	2nd Run	3rd Run	4th Run	5th Run	6th Run	7th Run	8th Run	9th Run	10th Run
addItem	45781	51491	43033	48978	48391	44726	42768	44258	41098	40251	52816
enter	70240.4	76085	68999	69330	68989	68085	65872	78102	66389	61612	78941
placeBid	29659.6	36290	27894	28344	26990	26015	27429	35964	25138	24305	38227
paySeller	36917	43657	34662	35099	34365	33461	35128	44196	33714	30724	44164
winnerPick	40387.5	47734	38043	39877	31834	37834	35810	49273	38261	36490	48719

Table 2: Our approach transaction cost

To test the benchmark approach, we tested ten times at different times of the day and took the average. The results are summarized in Table 3. For the methods in the contract, all methods have the same average gas fees of our approach except the placebid function which have an average gas cost of 67650.2.

Main Methods	**Avg. Gas Transaction Cost**	**1st Run**	**2nd Run**	**3rd Run**	**4th Run**	**5th Run**	**6th Run**	**7th Run**	**8th Run**	**9th Run**	**10th Run**
addItem	45781	51491	43033	48978	48391	44726	42768	44258	41098	40251	52816
enter	70240.4	76085	68999	69330	68989	68085	65872	78102	66389	61612	78941
placeBid	67650.2	74673	60091	69377	66560	64432	64013	76545	64232	66565	70014
paySeller	36917	43657	34662	35099	34365	33461	35128	44196	33714	30724	44164
winnerPick	40387.5	47734	38043	39877	31834	37834	35810	49273	38261	36490	48719

Table 3: Benchmark transaction cost.

The results indicate that all methods have approximately the same transaction cost except for the 'placeBid' method .In the benchmark approach shows a higher transaction cost than our approach; our approach selects bids for multiple items simultaneously and gives a unique number to each item our approach simplifies the auction process and reduces the overall number of transactions needed as the front end manage the auction highest bidder and only uses the place bid method when the auction ends which leads to lower gas fees.

5 Conclusion and Future Work

Our project successfully developed a decentralized online auction system using blockchain with features such as user authentication, location storage, bid management, and lottery. Our project shows blockchain's potential in an efficient auction platform. While essential design requirements were achieved, future work will be:

- Make the website pay using different tokens:
  Supporting various cryptocurrencies or tokens as payment options makes the platform more accessible to users who prefer different digital currencies. This increases the usability of the auction system.

- Implement advanced authentication methods such as biometric authentication and multi-factor authentication (MFA):
  Strengthening authentication methods beyond basic username and password adds an extra layer of security to the platform. Biometric authentication and multi-factor authentication (MFA) provide a stronger authentication process.

- Giving the website domain and making it on production instead of locally:
  Moving the website from a local environment to a production environment with its domain name makes the website accessible to a huge number of bidders making it easier for users to find and access the auction system.

- Completing the shipping and tracking system by using the stored location:
  Integrating a shipping and tracking system enhances the end-to-end transaction experience for users. This improves user satisfaction from the start to the end of the bidding process.

- Create a mobile application for the auction system:
  Developing mobile applications extends the reach of the auction system. This expands the user base and adds value to the overall auction experience.

6 References

[1] Z. Shi, C. de Laat, P. Grosso and Z. Zhao, "Integration of Blockchain and Auction Models: A Survey, Some Applications, and Challenges," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 497-537, Firstquarter 2023, doi: 10.1109/COMST.2022.3222403.

[2] Ilhaam A. Omar, Haya R. Hasan, Raja Jayaraman, Khaled Salah, Mohammed Omar, Implementing decentralized auctions using blockchain smart contracts, Technological Forecasting and Social Change, Volume 168, 2021, 120786, ISSN 0040-1625, https://doi.org/10.1016/j.techfore.2021.120786.

[3] Light, R. (2007), The Gambling Act 2005: Regulatory Containment and Market Control. The Modern Law Review, 70: 626-653. https://doi.org/10.1111/j.1468-2230.2007.00655.x

[4] Luo, Long & Feng, Jingcui & Hongfang, Yu & Sun, Gang. (2021). Blockchain-Enabled Two-Way Auction Mechanism for Electricity Trading in Internet of Electric Vehicles. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2021.3082769.

[5] V. P. Ranganthan, R. Dantu, A. Paul, P. Mears and K. Morozov, "A Decentralized Marketplace Application on the Ethereum Blockchain," *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, Philadelphia, PA, USA, 2018, pp. 90-97, doi: 10.1109/CIC.2018.00023.

[6] N. K. B, N. N. S, B. V. Surulikumar, K. R and B. K. R, "E-auction using Blockchain Mechanism," 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICAECA56562.2023.10200208.

[7] S. Kathar, H. Hardel, Z. Alam, S. Phansalkar and R. Sajjan, "Auction System for Agricultural Trade Using Blockchain Technology: A Survey with Proposed Framework," *2021 IEEE Pune Section International Conference (PuneCon)*, Pune, India, 2021, pp. 1-10, doi: 10.1109/PuneCon52575.2021.9686543.

[8] Thio-ac, A.; Domingo, E.J.; Reyes, R.; Arago, N.; Jorda, R.J.; Velasco, J. (2019). "Development of a Secure and Private Electronic Procurement System based on Blockchain Implementation," International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, pp. 2626-2631, 2019. doi: 10.30534/ijatcse/2019/115852019.

[9] Li, H., and W. Xue. (2021). A blockchain-based sealed-bid e-auction scheme with smart contract and zero-knowledge proof. Security and Communication Networks, 2021, 1-10, doi: 10.1155/2021/5523394

[10] H. S. Galal and A. M. Youssef, "Verifiable sealed-bid auction on the Ethereum blockchain," in Proceedings of the 2018 Financial Cryptography, pp. 265–278, Springer, Nieuwpoort, Curaçao, March 2018.

[11] Y. Peng, Y. Gao, and J. X. Wu, "A privacy-preserving sealed-bid auction scheme based on block chains," Cyberspace Security, vol. 9, no. 8, pp. 1–7, 2018.

[12] J. Xiong, "Research on Anonymous Electronic Auction Protocol Based on Blockchain," Jinan University, Guangzhou, China, 2019.

[13] R. Yu, "Research on the Sealed-Bid Auction Scheme for Blockchain Based on Secure Comparison Protocols," Northwest A&F University, Xianyang, China, 2019.

[14] M. Knecht, "Mandala: A smart contract programming language," arXiv preprint arXiv:1911.11376, 2019.

[15] Liu, Y., Liu, J., Salles, M.A.V., Zhang, Z., Li, T., Hu, B., Henglein, F., & Lu, R. (2022). Building blocks of sharding blockchain systems: Concepts, approaches, and open problems. Computer Science Review, 46, 100513.

[16] eBay, https://www.ebay.com/. Accessed: Dec. 24, 2023.

[17] M. S. Nikhil Kumar, G. C. Akshatha, M. D. Bangre, M. Dhanush and C. Abhishek, "Decentralized used cars Bidding application using Ethereum," 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2021, pp. 1-7, doi: 10.1109/CSITSS54238.2021.9682892.

[18] P. Manimaran and R. Dhanalakshmi, "Blockchain-Based Smart Contract for E-Bidding System," 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 2019, pp. 55-59, doi: 10.1109/ICCT46177.2019.8969042.

[19] ISO/TC 307 - Blockchain and Distributed Ledger Technologies." ISO. Available: https://www.iso.org/committee/6266604.html. Accessed: Jan. 8, 2024.

[20] ISO/IEC 27000 family Information security management" ISO. Available: https://www.iso.org/standard/iso-iec- 27000-family. Accessed: Jan. 8, 2024.

[21] Ethereum. "ERC-20 Token Standard." Ethereum Developer Documentation. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/. Accessed: Jan. 8, 2024.

[22] W3C. "About W3C Standards." World Wide Web Consortium, [Online]. Available: https://www.w3.org/standards/about/

[23] "HTML Living Standard." Web Hypertext Application Technology Working Group, [Online]. Available: https://html.spec.whatwg.org/

[24] WordPress. "CSS Coding Standards." WordPress Developer, [Online]. Available: https://developer.wordpress.org/coding-standards/wordpress-coding-standards/css/

[25] M. Grigutytė, "What is Bcrypt and how it works?" NordVPN, Jun. 16, 2023. [Online]. Available: https://nordvpn.com/blog/what-is-bcrypt/.

[26] "MongoDB: The Developer Data Platform," MongoDB, 2024. [Online]. Available: https://www.mongodb.com/ .

[27] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang and X. Luo, "CReam: A Smart Contract Enabled Collusion-Resistant e-Auction," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 7, pp. 1687-1701, July 2019, doi: 10.1109/TIFS.2018.2883275.

[28] J. Martins et al., "Fostering Customer Bargaining and E-Procurement Through a Decentralized Marketplace on the Blockchain," in IEEE Transactions on Engineering Management, vol. 69, no. 3, pp. 810-824, June 2022, doi: 10.1109/TEM.2020.3021242.

[29] "ethers.js v6 Documentation," The ethers.js Community, 2023. [Online]. Available: https://docs.ethers.org/v6/

[30] React. (n.d.). React. [Online]. Available: https://react.dev/