VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

J Component Report

Programme : B.Tech(CSE)

Course Title : Information Security Management

Course Code : CSE3502

Slot : A1

Title: Blockchain based Auction System

Team Members: Subhrato Som | 19BCE1119

Vibheesh VS | 19BAI1142

Rishi Varma Vegiraju | 19BCE1297

Faculty: Dr. Ayesha Shaik **Sign:**

Date: 25/04/2022

Table of Content

Abstract

e-Auction improves the efficiency of bid transaction. However, the protection of bidders' privacy, transaction fairness and verifiability, transaction data security, high cost of third-party auction center, and other issues have attracted more attention. In this project we have developed an auction system that uses a blockchain technologies to perform secure bidding. For developing this system we used solidity language and developed in remix etheirum editor. We know that the internet is changing with all the web3 frameworks and also blockchains as quite secured as of now. A blockchain is a distributed database that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Blockchains are best known for their crucial role in cryptocurrency systems, such as Bitcoin, for maintaining a secure and decentralized record of transactions.

Keywords: Beneficiary, Bid, AuctionEnd, Withdraw, highestBid, highestBidder.

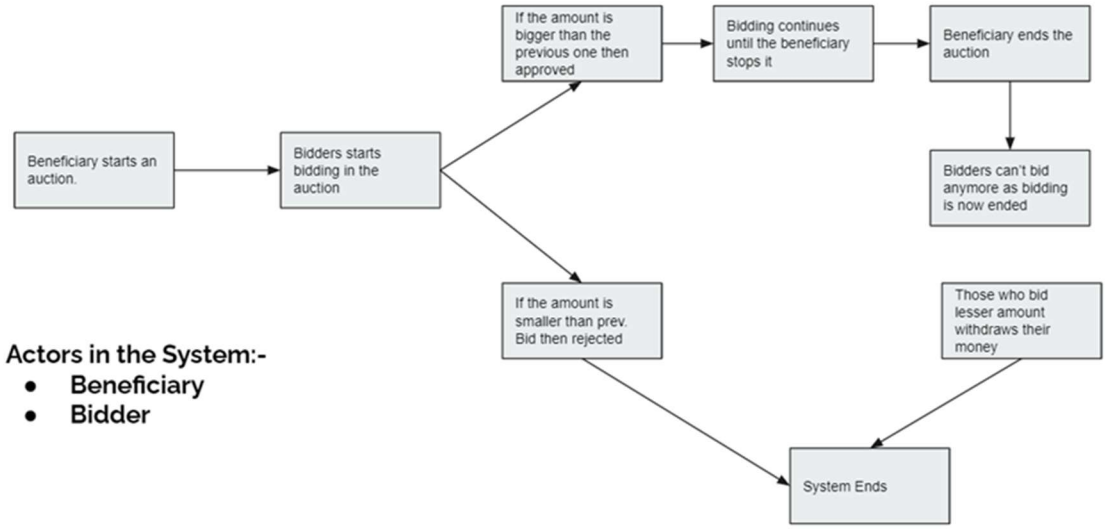
Introduction

In this project titled “Blockchain based Auction System”, we build an interface for the user to bid for the product by participating in the auction. For testing our auction system, we used ganache (a local Ethereum blockchain testing environment) for testing our smart contracts.
We also compiled our system in remix environment, similarly we compiled the code in local environment using truffle and then we have to start ganache in our local machine
Certain requirements for building our blochain based auction System are Solidity, Truffle, JavaScript, NPM, Remix, CLI.

Features of Our Auction System

- We can select any Bitcoin Address as our beneficiary.
- Once the bidding completes, the amount is deducted from the Highest bidder and that is added to the beneficiary account.
- The Previous Bid value is refunded automatically to the bidder bitcoin address if there is higher successive bid.
- If the successive bidder bids for product at lower value than the previous bid value, then his bid will not be added.
- The Gas value is added to transaction no matter the success or failure of the transaction.
- Even for the bidder fails to win the auction, Though, his amount or bid value is refunded back, but the Gas fees will not be refunded, it is used as price for processing the transaction.

Flow Diagram



Motivation

The efficiency of bid transactions is improved with e-Auction. However, issues like as bidder privacy, transaction fairness and verifiability, transaction data security, and the expensive expense of a third-party auction tools have gotten increasing attention. We investigated the challenges in current e-auction schemes using the transaction process and basic concepts of the sealed auction. We presented a e-auction scheme with smart contract technology based on Blockchain technology.

Requirements

- Remix Editor
- JavaScript
- Ganache
- Truffle
- Hyper

Literature Survey

[1] S. S. Sambare, N. Khandelwal, M. Nathwani, P. Munot and S. Patil, "A Survey of E-bidding System using Blockchain," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 250-255, doi: 10.1109/ICSSIT53264.2022.9716443.

With the increasing use of the internet, the e-auction system is becoming popular. E-Auction is an efficient e-commerce system that allows buyers and sellers to make a transaction through online platforms. Blockchain technology and smart contracts are being used in an e-auction to make it secure, transparent, and decentralized In this paper, we use different bidding techniques that use cryptography to ensure security and to keep private information confidential of any participants in the auction. Furthermore, we have explained the concept of public bidding and sealed bidding which will be used in E-auction, we have explained UAE that is used to keep stake holders identity autonomous, the paper also explains the hash functions which provides the advantage of encryption and decentralization Thus blockchain and smart contracts make it more efficient and secure compared to other traditional and e-auction systems.

[2] Solomon Antony a, Zhangxi Lin b, Bo Xu b, Determinants of escrow service adoption in consumer-to-consumer online auction market: An experimental study, Science-direct-2016.

This study also finds that sellers' reputation has a significant effect on buyer's risk perception, which influences his OES adoption decision. Furthermore, the buyers' OES adoption decisions were found to be congruent with the implied recommendations that were based on expected utility calculations. The data was collected from experimental C2C auction system with embedded decision support features. Results show that market factors, such as fraud rate, product price, and seller's reputation are important in determining buyers' OES adoption.

[3] Lafourcade P., Nopere M., Picot J., Pizzuti D., Roudeix E. (2020) Security Analysis of Auctionity: A Blockchain Based E-Auction. In:

Benzekri A., Barbeau M., Gong G., Laborde R., Garcia-Alfaro J. (eds) Foundations and Practice of Security. FPS 2019. Lecture Notes in Computer Science, vol 12056. Springer, Cham. https://doi.org/10.1007/978-3-030-45371-8_18

Auctions are widely used to sell products between different users. In this paper, they presented Auctionity, an English e-auction based on blockchain where they describe the different protocols used in Auctionity. they also define the security models and the associated properties. We formally prove some security properties of this protocol using ProVerif.

[4] L Ismanto, H Suwito Ar, A N Fajar, Sfenrianto and S Bachtiar, Blockchain as E-Commerce Platform in Indonesia, Journal of Physics: Conference Series, Volume 1179, 2019 J. Phys.: Conf. Ser. 1179 012114

The progress of technology and information system force digitalization process such as e-commerce to emerge. the utilization of e-commerce has flourished because it has advantages to broaden the market and stay competitive. However, it is faced with many challenges such as fraud, commission fees, limited contact between buyer and seller and misuse of personal data. Blockchain implementation has the potential to solve this problem with increased security and transparency through the implementation of cryptocurrency in payment and smart contracts. This paper explores the usage of the blockchain, cryptocurrency and smart contract to e-commerce for a secure and
efficient transaction in Indonesia. The result will be used to propose blockchain technology as e-commerce platform architecture and systems in Indonesia.

[5] Razan Aldaej, Latifa Alfowzan, Reem Alhashem, Mutasem K. Alsmadi, Ibrahim Al-Marashdeh, Analyzing, Designing and Implementing a Web-Based Auction online System, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 10 (2018) .

In the proposed OAS, the UML offering several diagrams to enable the new functions to be updated and added easily such as use case, sequence and class diagrams, and user interfaces. The proposed OAS will help the bidders to bid in fast and increase his chances to make a successful bid by suggesting a bid price, and help the seller to achieve maximum profit. Along with the tools that have been

used based on the analysis and implementation environment, the proposed OAS offers excellent advantages for the support of system development.

[6] Aaliya Sarfaraz,Ripon K. Chakrabortty, Daryl L. Essam, A tree structure-based improved blockchain framework for a secure online bidding system, 2021.

Human resources are essential in e-auctions, which frequently necessitate the use of a third-party middleman.This results in a considerable financial and time expense, and there is no guarantee that the third party is reliable. To address these concerns, this study provides a BC-based architecture for an open-bid auction system, in which multiple cryptographic primitives are used to consider privacy and security restrictions. This framework is unique in that it uses an improved strategy for combining BC structures by replacing the original chain structure with a tree structure.As a result, this paper suggests a cryptographic protocol for an open-bid auction on top of BC. The major goal is to create more efficiency in terms of security and privacy, which is mostly dependent on the performance of BC's encryption methods.

[7] Falade Adesola, Isaac Odun-Ayo, Moses Emetere, The Design and implementation of a secure online sealed-bid auction system, 2019

E-commerce has become one of the most extensively used methods for doing various types of business. Unfortunately, evil individuals have taken advantage of its diversity and appeal in a variety of ways. Customers of online auction platforms are increasingly becoming victims of fraudulent activities committed by these criminals. The backbone of this article is cryptography, which is used to illustrate and explain a practical implementation of a secure sealed-bid auction system. A variety of possible auction mechanisms are also addressed, which sparked the inspiration for this effort.

[8] Nazia Majadi,Jarrod Trevathan,Neil Bergmann, uAuction: Analysis, Design and Implementation of a Secure Online Auction System ,2016

Online auctions have become a hugely popular part of the online economy. During an auction, however, several fraudulent purchasing and selling behaviours might occur (e.g., shill bidding, bid shielding, etc. While experts are proposing techniques for countering such fraud, determining how effective these

countermeasures are is extremely difficult. This is owing to the fact that engaging in deceptive behaviour just for the sake of testing countermeasures is unethical. Furthermore, due to the sensitivity of an online auctioneer admitting that fraud has occurred or is occurring, there is limited commercial auction data available. We established our own online auction server for performing auction-related research in order to evaluate fraud countermeasures in a controlled setting.This paper describes our experiences creating and deploying uAuction, our own online auction system. There is
currently a scarcity of valuable material on auction system design. We utilise UML to display the architectural model, subsystems, use cases, activity workflows, class diagram, user interfaces, and system sequence diagrams in order to analyse and build the auction system. Our auction model is based on object-oriented principles and is open source, allowing other academics to improve it.

[9] Hani Qusa, Jumana Tarazi, Vishwesh Akre ,Secure E-Auction System Using Blockchain: UAE Case Study, 2020 Advances in Science and Engineering Technology International Conferences (ASET)©2020 IEEE

In this paper, they provided a prototype of secure blockchain e-auction system that lowering the uncertainties about identities of long-distance complex trade in an e-auction system that can be implemented in UAE services, especially, UAE Auction. In our implementation, we use smart contract in-order to guarantee the necessary security requirements. The smart contract contains important information about the transaction details such as auctioneer data, the start time and the deadline of auction, the current winner data, and the current highest price.

[10] Ilhaam A. Omar a , Haya R. Hasan b , Raja Jayaraman a,* , Khaled Salah b , Mohammed Omar a, Implementing decentralized auctions using blockchain smart contracts, 2021,120786,ISSN 0040-1625,Science-direct articles

In this paper, they propose a general framework for decentralized auctions leveraging (i) Ethereum smart contracts to trace and track bids, (ii) decentralized storage systems to upload documents related to bidding and (iii) trusted timer oracles that act as gateway between smart contract and external data feeds. In the pro posed solution, we develop detailed algorithms that define the working principles of the Smart contract for the auction process. We present detailed cost

analysis of the solution to demonstrate economic feasibility, providing a secure, transparent and reliable approach to online auctions.

Proposed Method

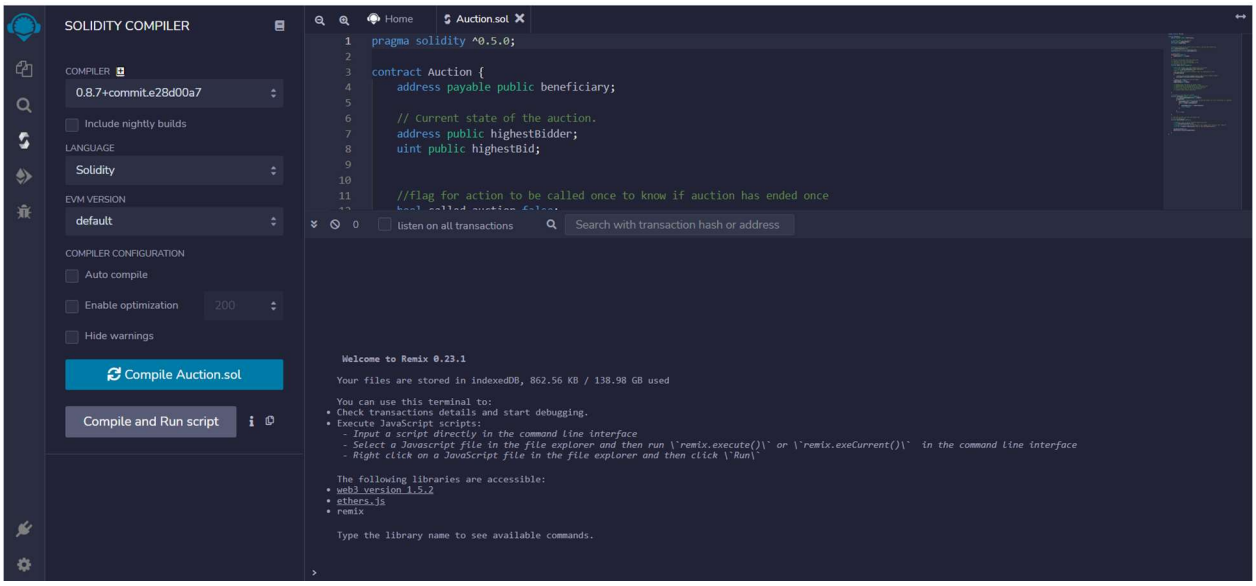
In this project we have used solidity language to develop a Blockchain based Auction System. In our proposed method there are two actors one is beneficiary and the other one is the bidder. The beneficiary is the one who starts the auctions and the bidders bid in that auction. There are 6 main functions with respect to the projects that are:-

- **beneficiary**(to set an account as an beneficiary).
- **bid(**it is for other accounts to bid in an auctions set by beneficiary).
- **auctionEnd**(it for ending the auction from beneficiary's side).
- **highestBidder**(this function gives tho output of the highest bidder in the system).
- **highestbid**(it is the highest bidding amount.
- **withdraw**(this function can be used by other bidder to take their wei back)

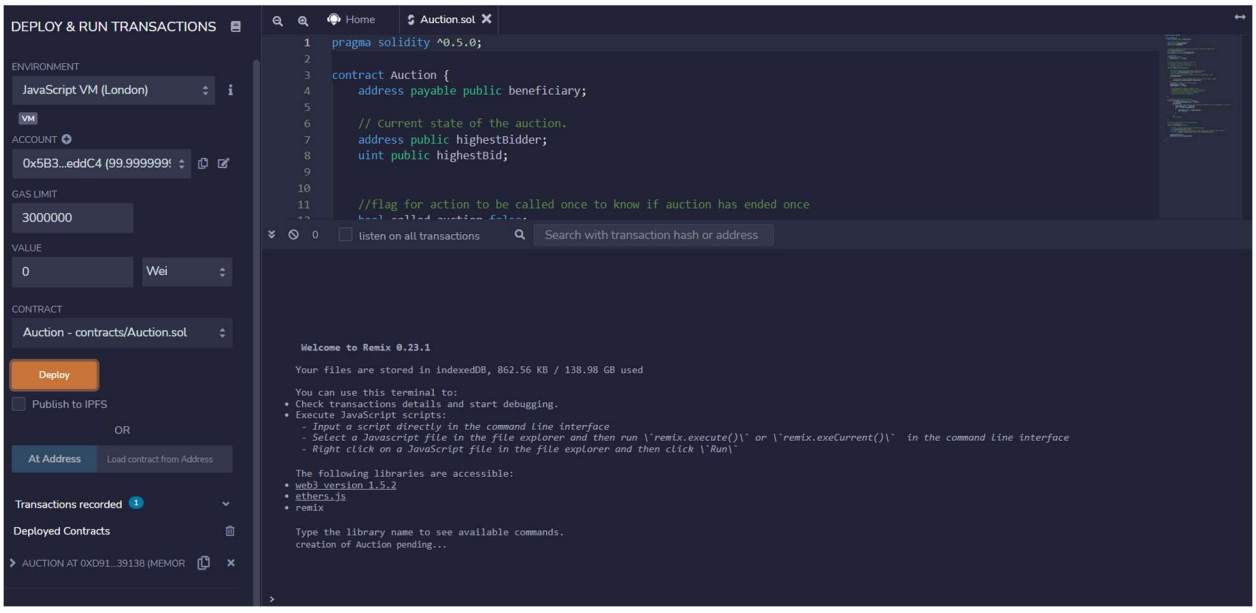
Implementation

Implementation in Remix

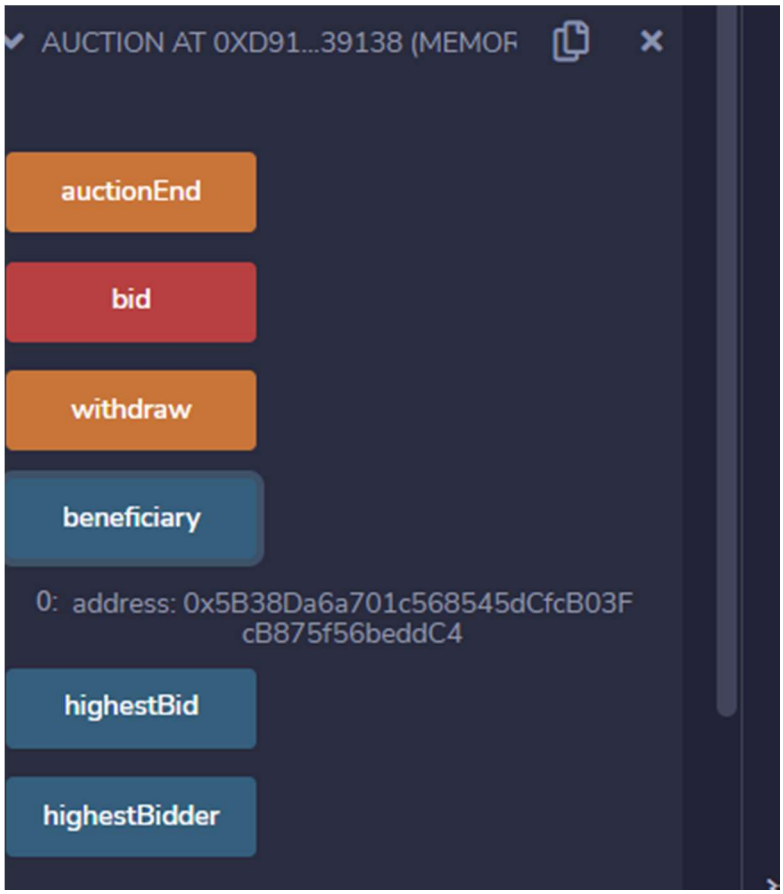
Compiling in remix editor



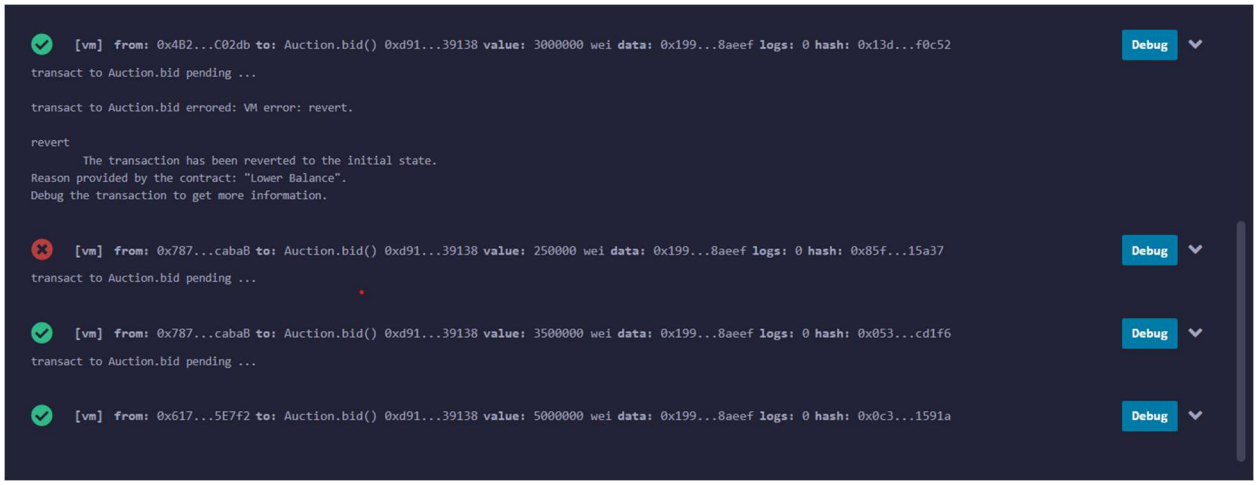
Deploying through first account



Setting the 0th account as beneficiary account

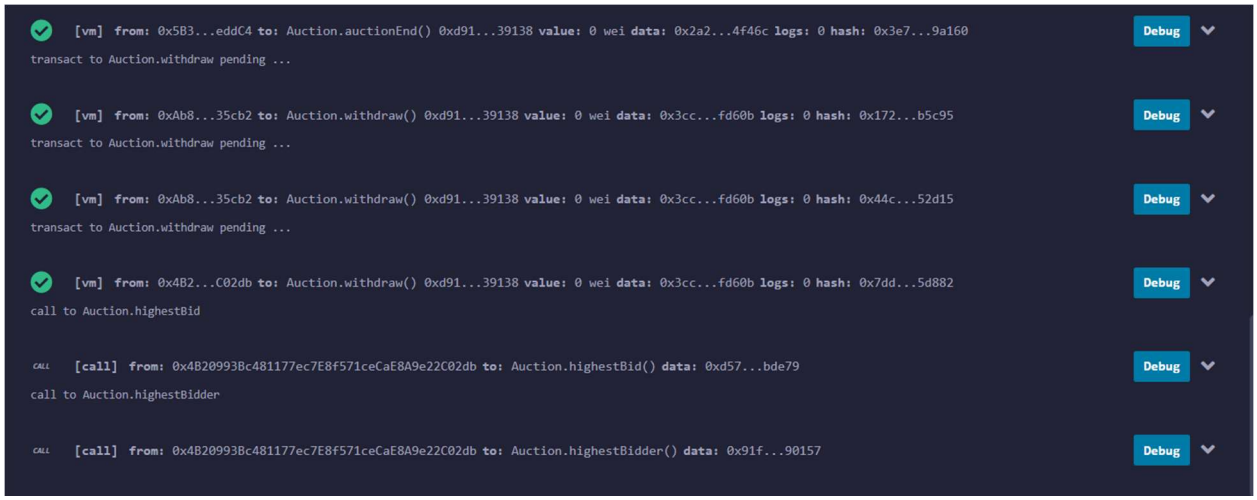


The Process of Bidding

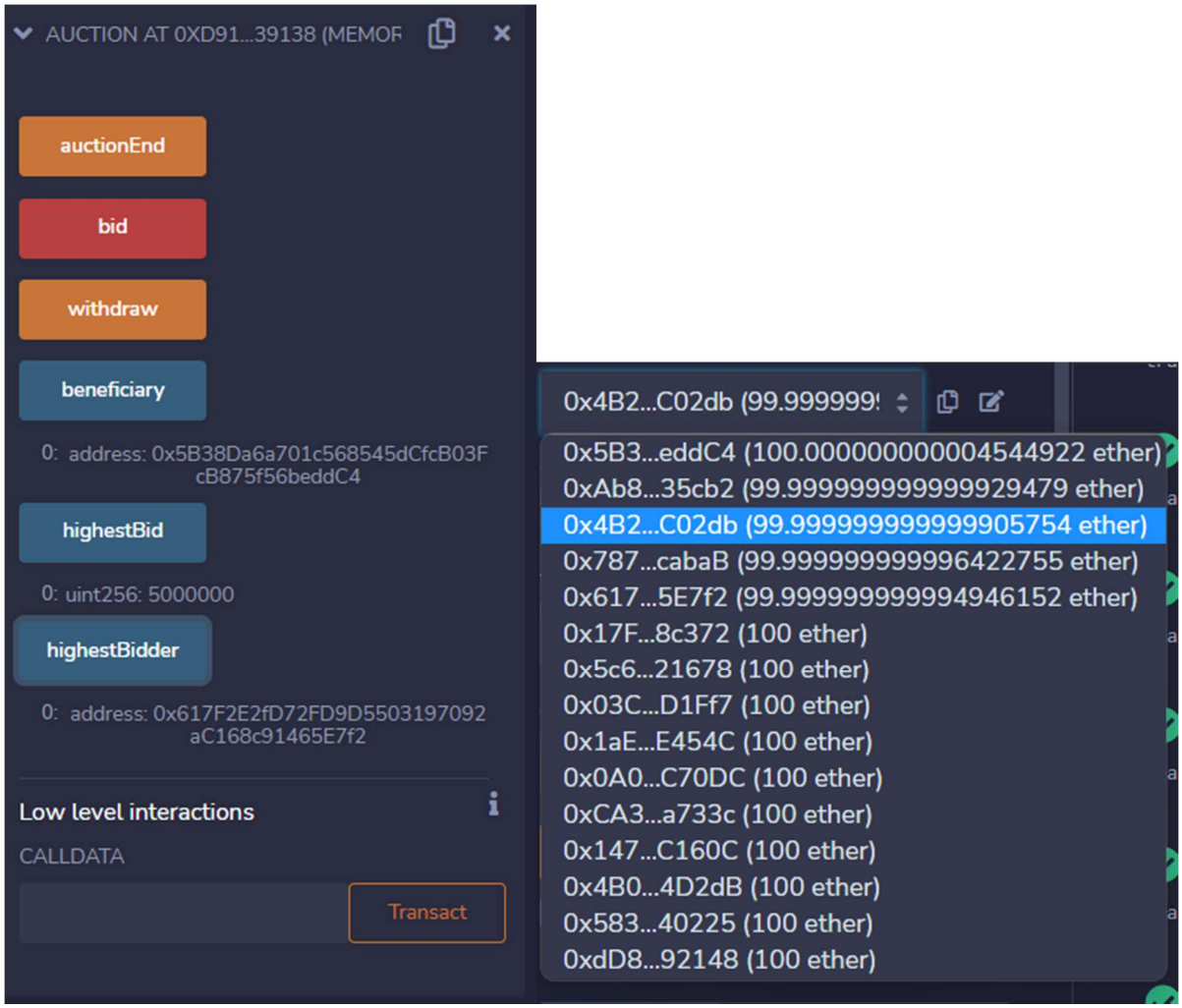


After the bidding is done the beneficiary can end the auction and the other bidders can

take there wei back but the reduced gas limit will not be returned



highestbid and highest bidder



Implementing the same system in local environment using truffle

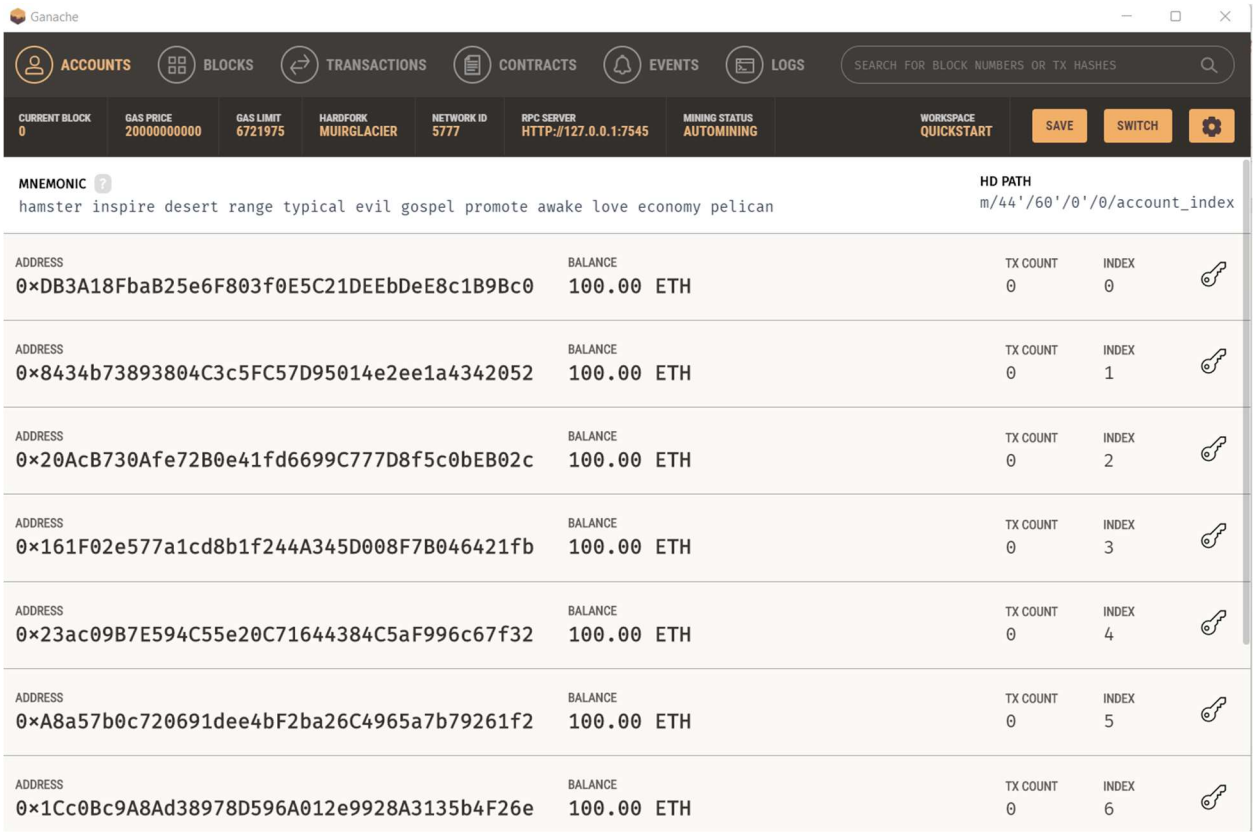
As we compiled in remix environment similarly we have to compile the code in local environment using truffle and then we have to start ganache in our local machine

```
subhr@LenovoIdeaPad MINGW64 ~/OneDrive/Desktop/Auctions
$ truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\Auction.sol
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\SimpleStorage.sol
> Artifacts written to C:\Users\subhr\OneDrive\Desktop\Auctions\client\src\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

subhr@LenovoIdeaPad MINGW64 ~/OneDrive/Desktop/Auctions
$
```

Ganache Interface



After that we need to deploy that in our system by migrating solidity code to javascript and data to json format. (Similar to remix deploying smart contracts here also requires gas value)

```
$ truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\Auction.sol
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\SimpleStorage.sol
> Artifacts written to C:\Users\subhr\OneDrive\Desktop\Blockchain based Auction System\c
lient\src\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


Starting migrations...
======================
> Network name:    'ganache'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0xf204c89ef6f7e1554e5ab295a2d3e1c52effadc90edc59b3a61f74e76783
0cc6
   > Blocks: 0            Seconds: 0
   > contract address:    0x533bb7a2B095c519cD6AD5863A89Df055B393057
   > block number:        1
   > block timestamp:     1650472156
   > account:             0xDB3A18FbaB25e6F803f0E5C21DEEbDeE8c1B9Bc0
   > balance:             99.9967165
   > gas used:            164175 (0x2814f)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0032835 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.0032835 ETH


2_deploy_contracts.js
=====================
```

```
2_deploy_contracts.js
=====================

   Replacing 'SimpleStorage'
   -------------------------
   > transaction hash:    0xbe1cf288a755549dc5560778c0e0a663c4ca3d666771a4ecb004ccf2fef6
a81b
   > Blocks: 0            Seconds: 0
   > contract address:    0x89ecdc9eD63e29f8B9Ca9123d16eD76Cb38043D6
   > block number:        3
   > block timestamp:     1650472158
   > account:             0xDB3A18FbaB25e6F803f0E5C21DEEbDeE8c1B9Bc0
   > balance:             99.9939459
   > gas used:            96189 (0x177bd)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00192378 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00192378 ETH

Summary
=======
> Total deployments:   2
> Final cost:          0.00520728 ETH
```

ADDRESS	BALANCE	TX COUNT	INDEX
0×DB3A18FbaB25e6F803f0E5C21DEEbDeE8c1B9Bc0	99.99 ETH	4	0

After that we are good to go and work with our console

```
subhr@LenovoIdeaPad MINGW64 ~/OneDrive/Desktop/Auctions
$ truffle console
truffle(ganache)>
```

Using all the functions by creating variable in the console

```
$ truffle console
truffle(ganache)> const instance = await Auction.deployed()
undefined
truffle(ganache)> instance
TruffleContract {
  constructor: [Function: TruffleContract] {
    _constructorMethods: {
      configureNetwork: [Function: configureNetwork],
      setProvider: [Function: setProvider],
      new: [Function: new],
      at: [AsyncFunction: at],
      deployed: [AsyncFunction: deployed],
      defaults: [Function: defaults],
      hasNetwork: [Function: hasNetwork],
      isDeployed: [Function: isDeployed],
      detectNetwork: [AsyncFunction: detectNetwork],
      setNetwork: [Function: setNetwork],
      setNetworkType: [Function: setNetworkType],
      setWallet: [Function: setWallet],
      resetAddress: [Function: resetAddress],
      link: [Function: link],
      clone: [Function: clone],
      addProp: [Function: addProp],
      toJSON: [Function: toJSON],
      decodeLogs: [Function: decodeLogs]
    },
```

Then creating the array list for the account connected with ganache

```
truffle(ganache)> const arr_list = await web3.eth.getAccounts();
undefined
truffle(ganache)> arr_list
[
  '0xDB3A18FbaB25e6F803f0E5C21DEEbDeE8c1B9Bc0',
  '0x8434b73893804C3c5FC57D95014e2ee1a4342052',
  '0x20AcB730Afe72B0e41fd6699C777D8f5c0bEB02c',
  '0x161F02e577a1cd8b1f244A345D008F7B046421fb',
  '0x23ac09B7E594C55e20C71644384C5aF996c67f32',
  '0xA8a57b0c720691dee4bF2ba26C4965a7b79261f2',
  '0x1Cc0Bc9A8Ad38978D596A012e9928A3135b4F26e',
  '0xf2963E03C20e606b96506d97D682A3bBE2A2fA60',
  '0xfD0285F1ed358b463f3D4676cfff212e07E632f8',
  '0x183E2091d888754E760a512278ee905Dc01fd6ef'
]
truffle(ganache)>
```

Then creating the array list for the account connected with ganache

```
truffle(ganache)> const arr_list = await web3.eth.getAccounts();
undefined
truffle(ganache)> arr_list
[
  '0xDB3A18FbaB25e6F803f0E5C21DEEbDeE8c1B9Bc0',
  '0x8434b73893804C3c5FC57D95014e2ee1a4342052',
  '0x20AcB730Afe72B0e41fd6699C777D8f5c0bEB02c',
  '0x161F02e577a1cd8b1f244A345D008F7B046421fb',
  '0x23ac09B7E594C55e20C71644384C5aF996c67f32',
  '0xA8a57b0c720691dee4bF2ba26C4965a7b79261f2',
  '0x1Cc0Bc9A8Ad38978D596A012e9928A3135b4F26e',
  '0xf2963E03C20e606b96506d97D682A3bBE2A2fA60',
  '0xfD0285F1ed358b463f3D4676cfff212e07E632f8',
  '0x183E2091d888754E760a512278ee905Dc01fd6ef'
]
truffle(ganache)>
```

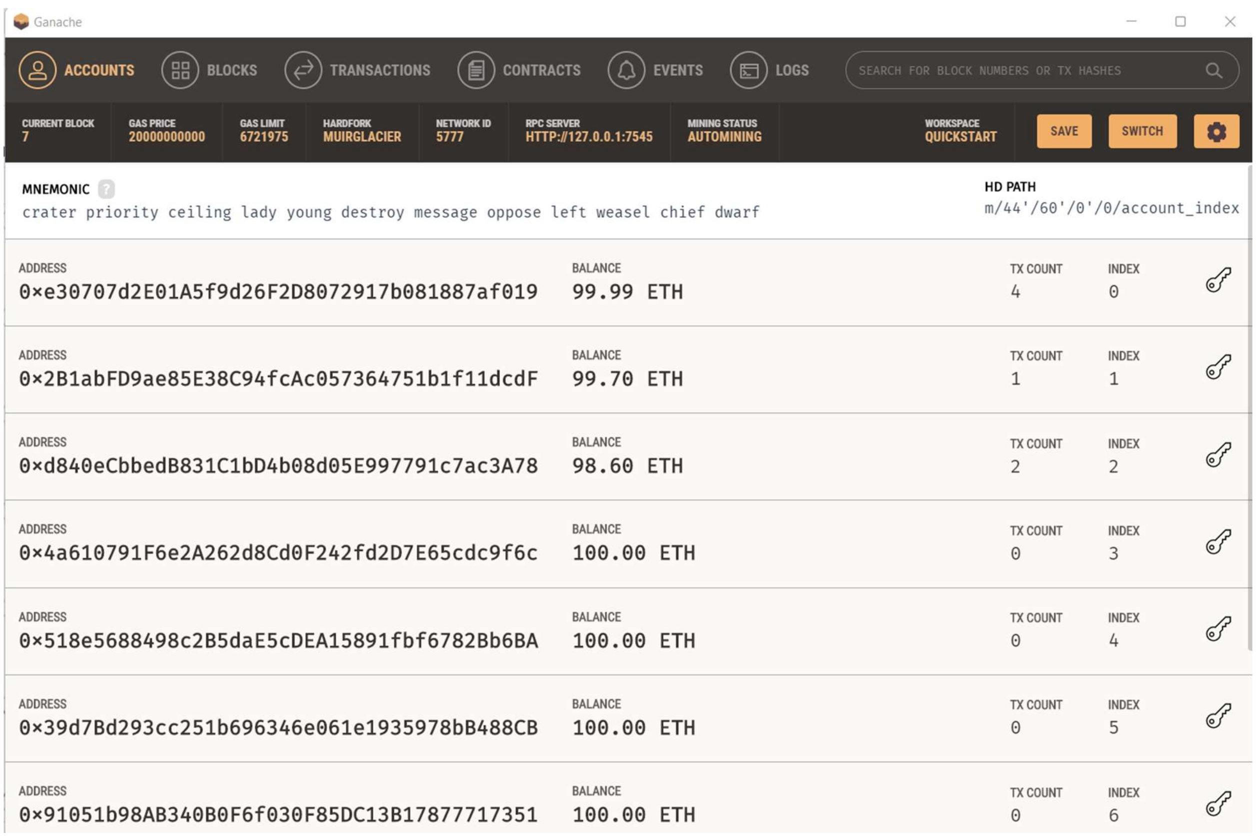
Making 0th account as beneficiary and creating first bid from 1st account

```
truffle(ganache)> const benf =arr_list[0];
undefined
truffle(ganache)> await instance.bid(from:arr_list[1], value:3*100000000);
evalmachine.<anonymous>:1
  await instance.bid(from:arr_list[1], value:3*100000000)
                     ^^^^

Uncaught SyntaxError: missing ) after argument list
truffle(ganache)> await instance.bid({from:arr_list[1], value:3*100000000});
{
  tx: '0x2e21e669082c0670319d21cfc814634ae63e85671748cea929c9f7d4f56391c8',
  receipt: {
    transactionHash: '0x2e21e669082c0670319d21cfc814634ae63e85671748cea929c9f7d4f56391c8',
    transactionIndex: 0,
    blockHash: '0xd0766a47ae95a8427c8fe950f7e331e195d41c5464413f667962dd8c173b710e',
    blockNumber: 9,
    from: '0x8434b73893804c3c5fc57d95014e2ee1a4342052',
    to: '0x3dbdcbf251b359ed0f7bf2230a7d36db41db6ac3',
    gasUsed: 63689,
    cumulativeGasUsed: 63689,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
```

If a bid is smaller than the previous one then it gives error

```
truffle(ganache)> await instance.bid({from:arr_list[2], value:9*100000000});
{
  tx: '0xe6697337543eeeaffbfcf15309f9eed65661105b128b12034d8dcfbcb466dca6',
  receipt: {
    transactionHash: '0xe6697337543eeeaffbfcf15309f9eed65661105b128b12034d8dcfbcb466dca6',
    transactionIndex: 0,
    blockHash: '0x7ef940ad390dec02fb6611d2d536655aae3da38fa6910c7855dbf7656f919ddf',
    blockNumber: 10,
    from: '0x20acb730afe72b0e41fd6699c777d8f5c0beb02c',
    to: '0x3dbdcbf251b359ed0f7bf2230a7d36db41db6ac3',
    gasUsed: 56248,
    cumulativeGasUsed: 56248,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> await instance.bid({from:arr_list[2], value:5*100000000});
Uncaught:
Error: Returned error: VM Exception while processing transaction: revert Lower Balance -- Reason given: Lower Bala
nce.
    at evalmachine.<anonymous>:1:18
```



Ending the auction through first account

```
truffle(ganache)> await instance.auctionEnd({from:accounts[0]});
{
  tx: '0x61ce756dab8d3d3624c9897a0d2d722e148e9f8bd518de6c76563b52f933cdad',
  receipt: {
    transactionHash: '0x61ce756dab8d3d3624c9897a0d2d722e148e9f8bd518de6c76563b52f933cdad',
    transactionIndex: 0,
    blockHash: '0x9e40d1de04784127ff2e42e60dd16e3adb909a4dd612c7b67a10dfdb0ec5d362',
    blockNumber: 8,
    from: '0xe30707d2e01a5f9d26f2d8072917b081887af019',
    to: '0x8594a7f30e12689ed750dacec8232f8769ffdcdb',
    gasUsed: 52956,
    cumulativeGasUsed: 52956,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)>
```

ADDRESS	BALANCE	TX COUNT	INDEX
0×e30707d2E01A5f9d26F2D8072917b081887af019	100.89 ETH	5	0
0×2B1abFD9ae85E38C94fcAc057364751b1f11dcdF	99.70 ETH	1	1
0×d840eCbbedB831C1bD4b08d05E997791c7ac3A78	98.60 ETH	2	2
0×4a610791F6e2A262d8Cd0F242fd2D7E65cdc9f6c	100.00 ETH	0	3
0×518e5688498c2B5daE5cDEA15891fbf6782Bb6BA	100.00 ETH	0	4
0×39d7Bd293cc251b696346e061e1935978bB488CB	100.00 ETH	0	5
0×91051b98AB340B0F6f030F85DC13B17877717351	100.00 ETH	0	6

```
truffle(ganache)> await instance.withdraw({from:accounts[1]});
{
  tx: '0xab1757343d02ed449d2608d5ee51877044a86767aadd525e2c7555fd0ed17812',
  receipt: {
    transactionHash: '0xab1757343d02ed449d2608d5ee51877044a86767aadd525e2c7555fd0ed17812',
    transactionIndex: 0,
    blockHash: '0x85630dddbe5f61187bcd212c67edb4df8f2b71a264d6e3c1e485125e07c9ea9a',
    blockNumber: 9,
    from: '0x2b1abfd9ae85e38c94fcac057364751b1f11dcdf',
    to: '0x8594a7f30e12689ed750dacec8232f8769ffdcdb',
    gasUsed: 19857,
    cumulativeGasUsed: 19857,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
```

As seen the amount getting deduced.

MNEMONIC

crater priority ceiling lady young destroy message oppose left weasel chief dwarf

HD PATH

m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×e30707d2E01A5f9d26F2D8072917b081887af019	100.89 ETH	5	0
0×2B1abFD9ae85E38C94fcAc057364751b1f11dcdF	100.00 ETH	2	1
0×d840eCbbedB831C1bD4b08d05E997791c7ac3A78	99.10 ETH	3	2
0×4a610791F6e2A262d8Cd0F242fd2D7E65cdc9f6c	100.00 ETH	0	3
0×518e5688498c2B5daE5cDEA15891fbf6782Bb6BA	100.00 ETH	0	4
0×39d7Bd293cc251b696346e061e1935978bB488CB	100.00 ETH	0	5
0×91051b98AB340B0F6f030F85DC13B17877717351	100.00 ETH	0	6

Highest bid and highest bidder

```
truffle(ganache)> await instance.highestBid();
BN {
  negative: 0,
  words: [ 31064064, 56381138, 199, <1 empty item> ],
  length: 3,
  red: null
}
truffle(ganache)> await instance.highestBidder();
'0xd840eCbbedB831C1bD4b08d05E997791c7ac3A78'
truffle(ganache)>
```

Code Link

https://drive.google.com/file/d/1QgGAG6z1Iu5UDJaOn1Y29NGl2jQHIxZ2/view?usp=sharing

Conclusion

Finally we would like to conclude that we have made a fully functional auction system which uses Blockchain to store all the data, which also makes it more secured. For obtaining we used solidity as our primary language and then extended it to java script using truffle library

References

- S. S. Sambare, N. Khandelwal, M. Nathwani, P. Munot and S. Patil, "A Survey of E-bidding System using Blockchain," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022.
- Solomon Antony a, Zhangxi Lin b, Bo Xu b, Determinants of escrow service adoption in consumer-to-consumer online auction market: An experimental study, Science-direct-2016.
- Lafourcade P., Nopere M., Picot J., Pizzuti D., Roudeix E. (2020) Security Analysis of Auctionity: A Blockchain Based E-Auction. In: Benzekri A., Barbeau M., Gong G., Laborde R., Garcia-Alfaro J. (eds) Foundations and Practice of Security. FPS 2019. Lecture Notes in Computer Science, vol 12056. Springer, Cham.
- L Ismanto, H Suwito Ar, A N Fajar, Sfenrianto and S Bachtiar, Blockchain as E-Commerce Platform in Indonesia, Journal of Physics: Conference Series, Volume 1179, 2019 J. Phys.: Conf. Ser. 1179 012114.
- Razan Aldaej, Latifa Alfowzan, Reem Alhashem, Mutasem K. Alsmadi, Ibrahim Al-Marashdeh, Analyzing, Designing and Implementing a Web-Based Auction online System, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 10 (2018) .
- Aaliya Sarfaraz,Ripon K. Chakrabortty, Daryl L. Essam, A tree structure-based improved blockchain framework for a secure online bidding system, 2021.
- Falade Adesola, Isaac Odun-Ayo, Moses Emetere, The Design and implementation of a secure online sealed-bid auction system, 2019.
- Nazia Majadi,Jarrod Trevathan,Neil Bergmann, uAuction: Analysis, Design and Implementation of a Secure Online Auction System ,2016.
- Hani Qusa, Jumana Tarazi, Vishwesh Akre ,Secure E-Auction System Using Blockchain: UAE Case Study, 2020 Advances in Science and Engineering Technology International Conferences (ASET)©2020 IEEE.
- Ilhaam A. Omar a , Haya R. Hasan b , Raja Jayaraman a,* , Khaled Salah b , Mohammed Omar a, Implementing decentralized auctions using blockchain smart contracts, 2021,120786,ISSN 0040-1625,Science-direct articles.