BLOCKCHAIN

HW 1 – REPORT

Name: NIKHILESH REDDY TUMMALA

UFID: 8350 – 1593

Email: tummalanikhilesh@ufl.edu

Changes Made in Auction.Sol:

In the constructor, we assign the beneficiary to the msg.sender and the highestbid is equated to zero. The Boolean variable "Auction_Terminated" is set to false. In the bid() function, we bid on the auction with the value sent together with this transaction. The value will only be refunded if the auction is not won. If the bid is not higher than the already obtained highest bid, then the money is sent back to the person who offered that corresponding bid. In the solidity code, we put it as "require(msg.value > highestBid);" which checks if the proposed bid is greater than the already achieved highest bid or not. If it isn't, then the money needs to be sent back to the person who proposed it, or else, we update the highest bid to the newly proposed bid and change the address from the previous bidder to the new one who proposed the corresponding high bid.

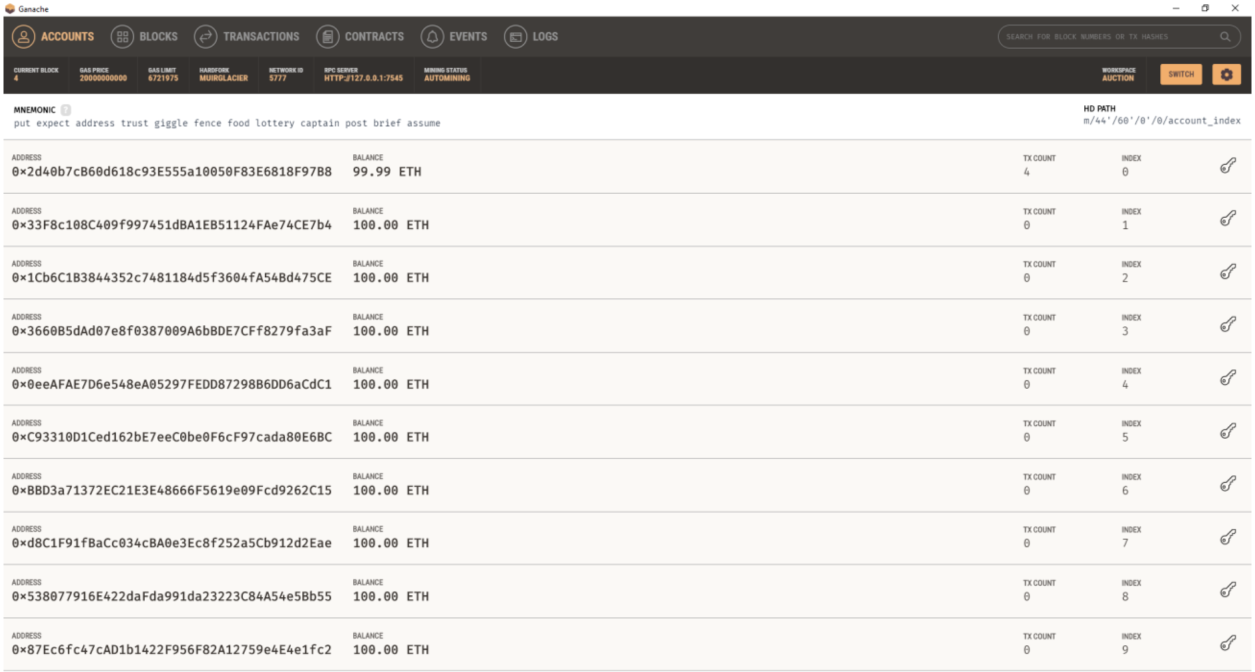
In the pendingResults map, we store the address of the user who's bid was previously considered as highest and their corresponding bid, so that we send these users their money back to them. If Anna bids for 10 ETH, Bob bids for 15 ETH, then the address of Anna and it's bid money which is 10 ETH will be stored in the pendingResults map, because before Bob gave his bid for 15 ETH, Anna was the highest bidder with 10 ETH and now we have to return those ETH back to her, which makes Bob the current highest bidder until someone else bids higher than him.

To avoid the reentrancy attack, the function first takes the amount that needs to be transferred to the person who called the withdraw() function, does the necessary update and then send the money. For the auctionEnd() function, firstly it make sures that apart from beneficiary, no one else calls the function and then checks, if the auction has ended or not. If it isn't then the Boolean variable "Auction_Terminated" is made true and the bid money is sent to the beneficiary account.

I have attached the screenshots from the truffle console and from ganache after each function call.

1. <u>Before and After images for functions bid(), withdraw() and auctionEnd():</u>

   a. <u>Before bid():</u> All the accounts here, have 100 ETH each. The user address whose index is 0, is our beneficiary account. Right now, no one has done the bidding and so, all of them have the same amount.



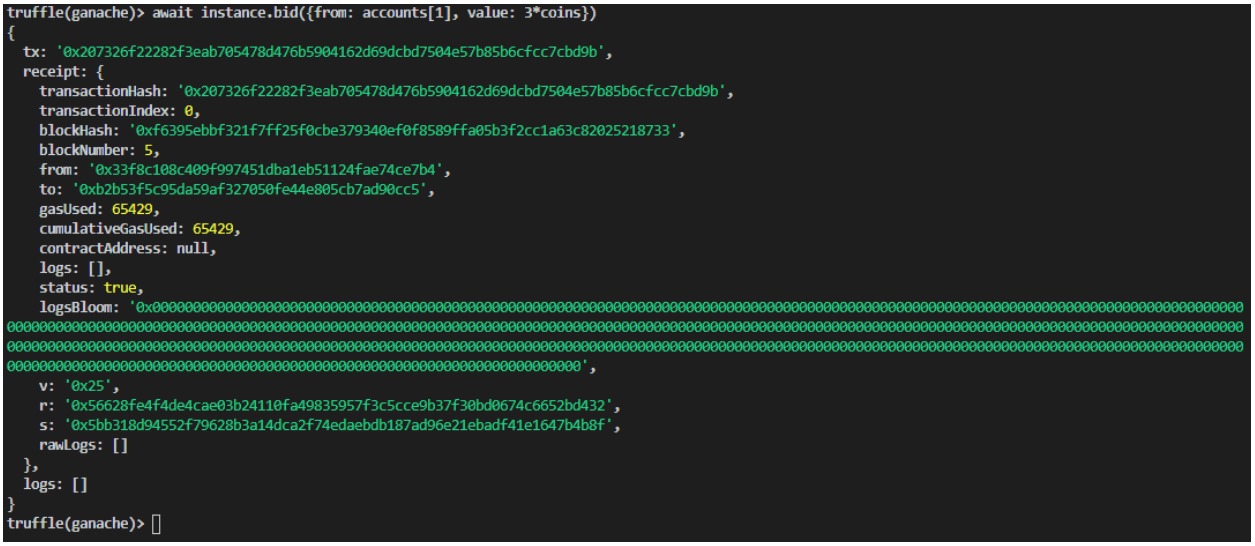
CURRENT BLOCK 4
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE AUCTION

MNEMONIC
put expect address trust giggle fence food lottery captain post brief assume

HD PATH
m/44'/60'/0'/0/account_index

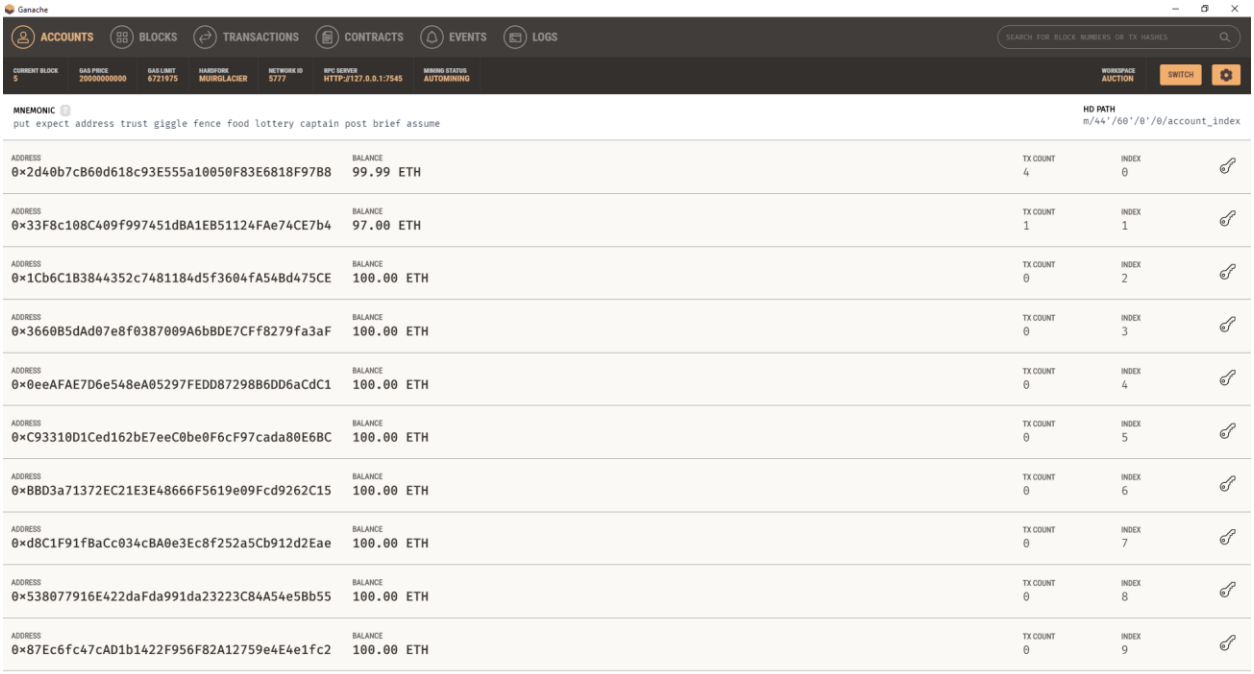
ADDRESS	BALANCE	TX COUNT	INDEX
0×2d40b7cB60d618c93E555a10050F83E6818F97B8	99.99 ETH	4	0
0×33F8c108C409f997451dBA1EB51124FAe74CE7b4	100.00 ETH	0	1
0×1Cb6C1B3844352c7481184d5f3604fA54Bd475CE	100.00 ETH	0	2
0×3660B5dAd07e8f0387009A6bBDE7CFf8279fa3aF	100.00 ETH	0	3
0×0eeAFAE7D6e548eA05297FEDD87298B6DD6aCdC1	100.00 ETH	0	4
0×C93310D1Ced162bE7eeC0be0F6cF97cada80E6BC	100.00 ETH	0	5
0×BBD3a71372EC21E3E48666F5619e09Fcd9262C15	100.00 ETH	0	6
0×d8C1F91fBaCc034cBA0e3Ec8f252a5Cb912d2Eae	100.00 ETH	0	7
0×538077916E422daFda991da23223C84A54e5Bb55	100.00 ETH	0	8
0×87Ec6fc47cAD1b1422F956F82A12759e4E4e1fc2	100.00 ETH	0	9

   b. <u>After bid():</u> After executing the below command, we get the output, “await instance.bid({from: accounts[1], value: 3*coins})”



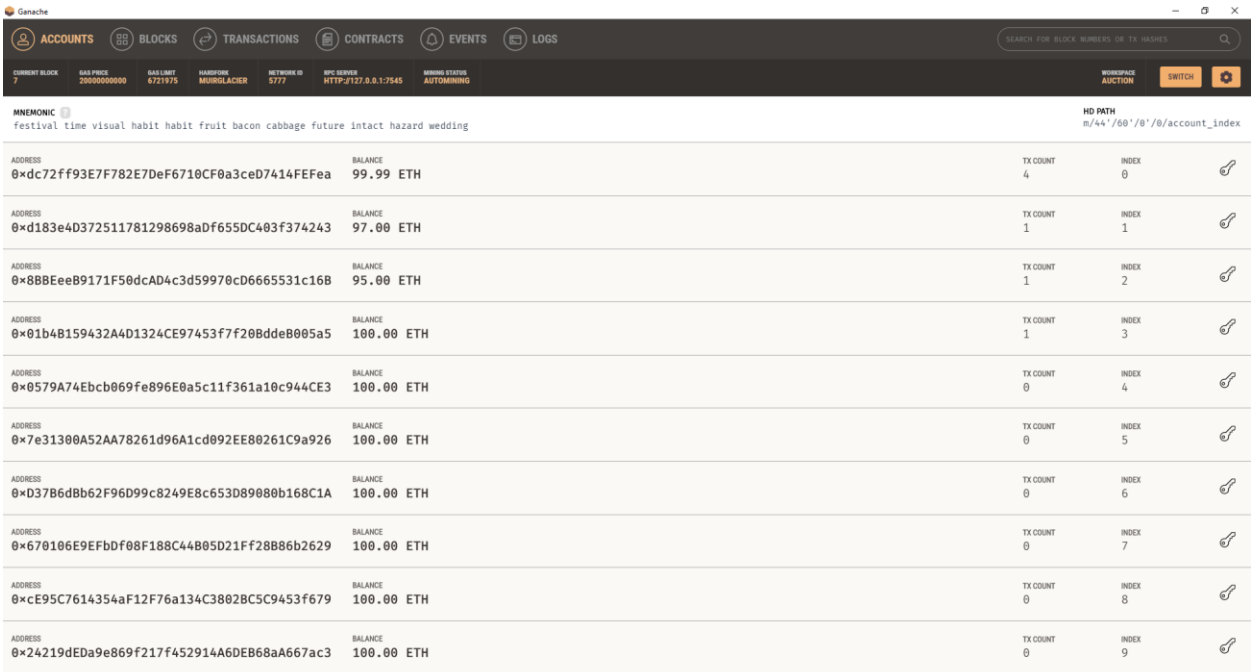
```
truffle(ganache)> await instance.bid({from: accounts[1], value: 3*coins})
{
  tx: '0x207326f22282f3eab705478d476b5904162d69dcbd7504e57b85b6cfcc7cbd9b',
  receipt: {
    transactionHash: '0x207326f22282f3eab705478d476b5904162d69dcbd7504e57b85b6cfcc7cbd9b',
    transactionIndex: 0,
    blockHash: '0xf6395ebbf321f7ff25f0cbe379340ef0f8589ffa05b3f2cc1a63c82025218733',
    blockNumber: 5,
    from: '0x33f8c108c409f997451dba1eb51124fae74ce7b4',
    to: '0xb2b53f5c95da59af327050fe44e805cb7ad90cc5',
    gasUsed: 65429,
    cumulativeGasUsed: 65429,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000',
    v: '0x25',
    r: '0x56628fe4f4de4cae03b24110fa49835957f3c5cce9b37f30bd0674c6652bd432',
    s: '0x5bb318d94552f79628b3a14dca2f74edaebdb187ad96e21ebadf41e1647b4b8f',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> 
```

And this reflects in Ganache as below:

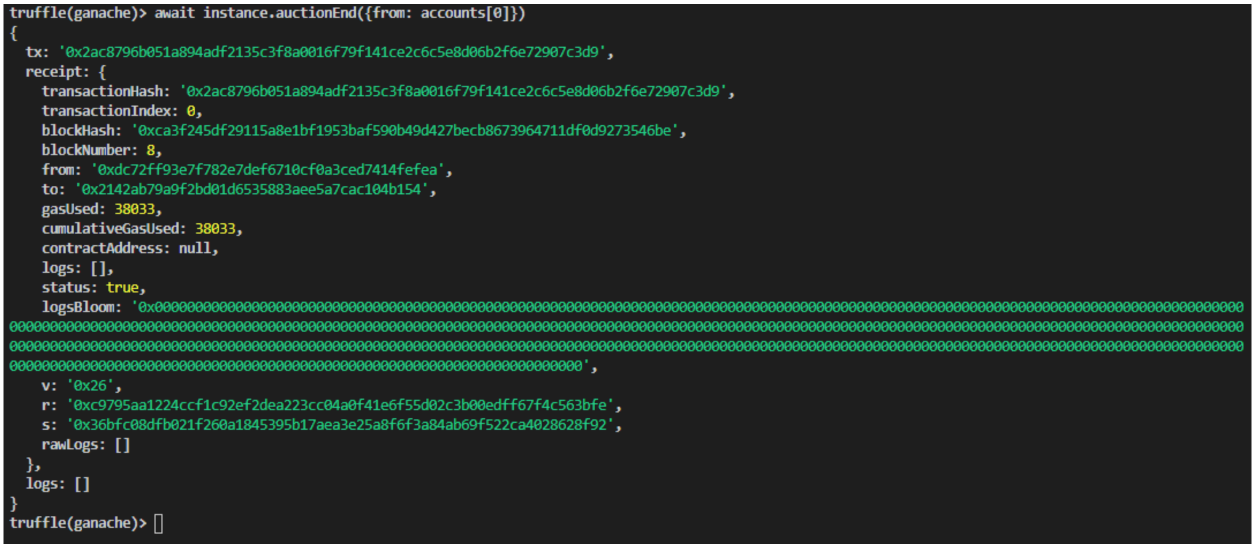


Here we can see that the user with index 1 has 3 ETH less compared to others and this is because he has bid 3 ETH and so the same amount has been reduced from him. He will get this money back is anyone else bids more than him which is more than 3 ETH.

c. <u>Before auctionEnd():</u> Before this function is called

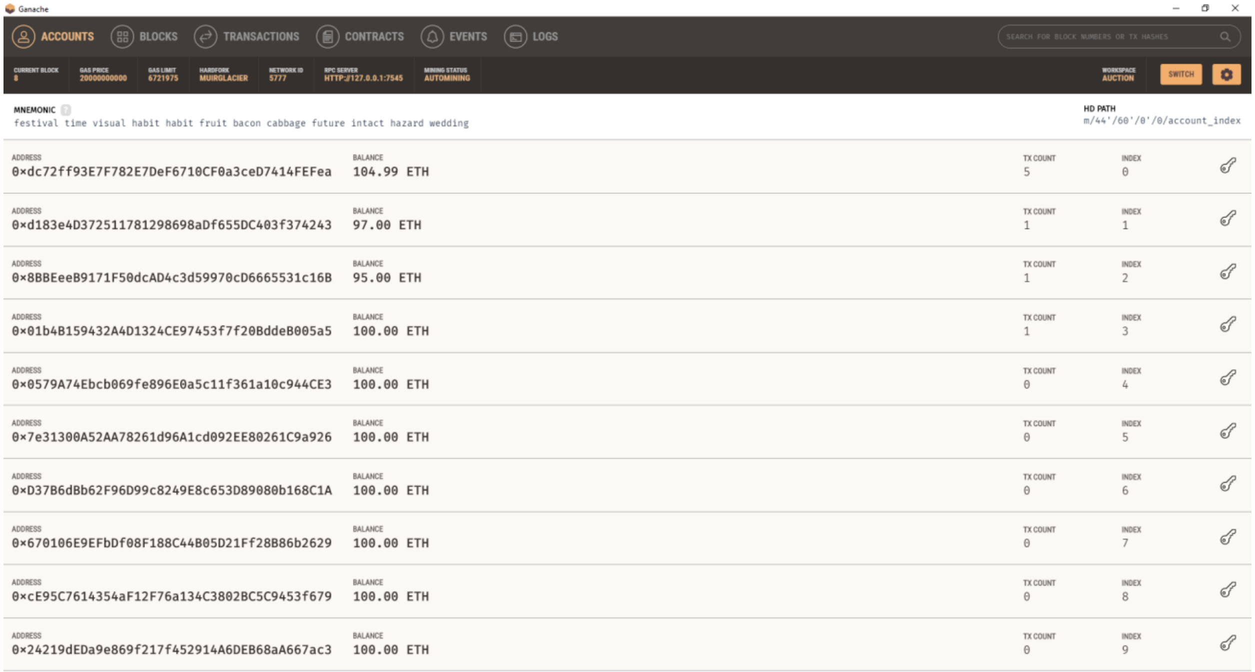


d. **After auctionEnd():** After auctionEnd() function ran on the truffle console, below is the output shown.



```
truffle(ganache)> await instance.auctionEnd({from: accounts[0]})
{
  tx: '0x2ac8796b051a894adf2135c3f8a0016f79f141ce2c6c5e8d06b2f6e72907c3d9',
  receipt: {
    transactionHash: '0x2ac8796b051a894adf2135c3f8a0016f79f141ce2c6c5e8d06b2f6e72907c3d9',
    transactionIndex: 0,
    blockHash: '0xca3f245df29115a8e1bf1953baf590b49d427becb8673964711df0d9273546be',
    blockNumber: 8,
    from: '0xdc72ff93e7f782e7def6710cf0a3ced7414fefea',
    to: '0x2142ab79a9f2bd01d6535883aee5a7cac104b154',
    gasUsed: 38033,
    cumulativeGasUsed: 38033,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000',
    v: '0x26',
    r: '0xc9795aa1224ccf1c92ef2dea223cc04a0f41e6f55d02c3b00edff67f4c563bfe',
    s: '0x36bfc08dfb021f260a1845395b17aea3e25a8f6f3a84ab69f522ca4028628f92',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> 
```

If we notice, the user with index 0 (which is our beneficiary account) received an additional 5 ETH from the user with index 2 who have 5 ETH less than 100 ETH. This user has bid the highest which is 5 ETH compared to the others and so the same amount that he bid, was received by the beneficiary.



CURRENT BLOCK 8
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE AUCTION

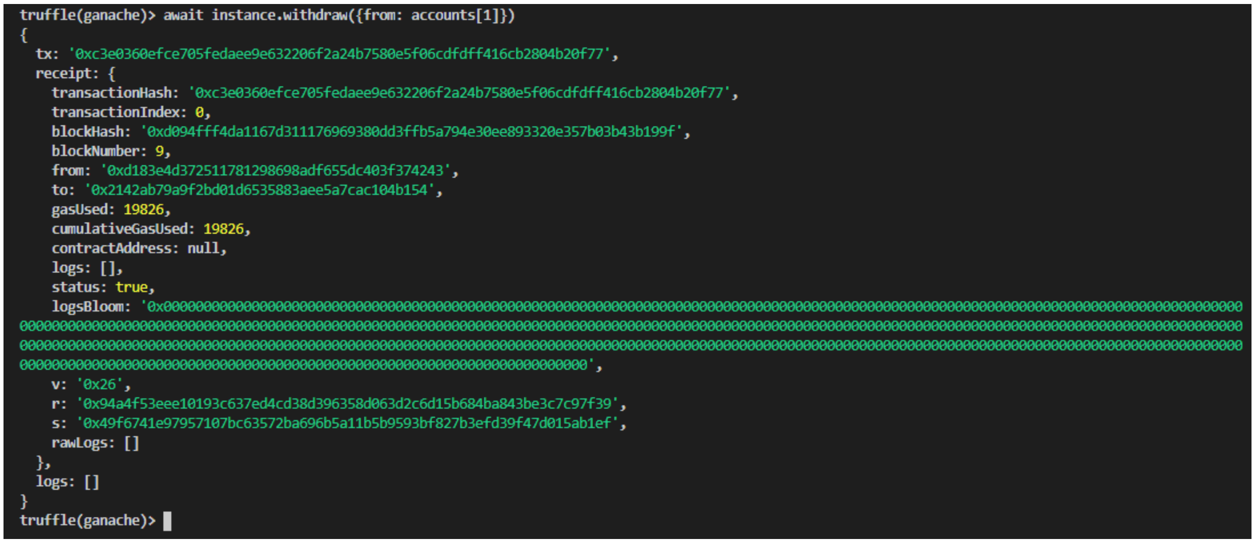
MNEMONIC
festival time visual habit habit fruit bacon cabbage future intact hazard wedding

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×dc72ff93E7F782E7DeF6710CF0a3ceD7414FEFea	104.99 ETH	5	0
0×d183e4D372511781298698aDf655DC403f374243	97.00 ETH	1	1
0×8BBEeeB9171F50dcAD4c3d59970cD6665531c16B	95.00 ETH	1	2
0×01b4B159432A4D1324CE97453f7f20BddeB005a5	100.00 ETH	1	3
0×0579A74Ebcb069fe896E0a5c11f361a10c944CE3	100.00 ETH	0	4
0×7e31300A52AA78261d96A1cd092EE80261C9a926	100.00 ETH	0	5
0×D37B6dBb62F96D99c8249E8c653D89080b168C1A	100.00 ETH	0	6
0×670106E9EFbDf08F188C44B05D21Ff28B86b2629	100.00 ETH	0	7
0×cE95C7614354aF12F76a134C3802BC5C9453f679	100.00 ETH	0	8
0×24219dEDa9e869f217f452914A6DEB68aA667ac3	100.00 ETH	0	9

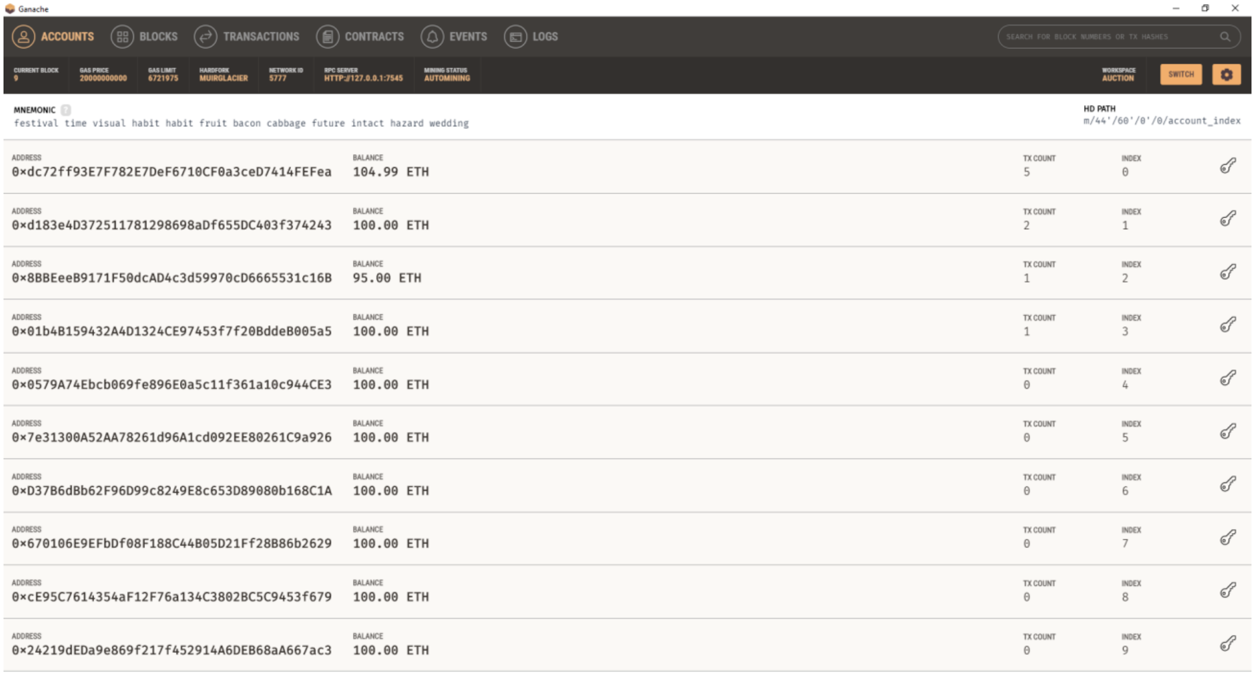
e. **Before withdraw ():** The picture above, shows that the other users who bid less than the highest are still yet to receive their bid amount.

f. **After withdraw ():** Below is the output from the truffle console when the withdraw() is executed.



```
truffle(ganache)> await instance.withdraw({from: accounts[1]})
{
  tx: '0xc3e0360efce705fedaee9e632206f2a24b7580e5f06cdfdff416cb2804b20f77',
  receipt: {
    transactionHash: '0xc3e0360efce705fedaee9e632206f2a24b7580e5f06cdfdff416cb2804b20f77',
    transactionIndex: 0,
    blockHash: '0xd094fff4da1167d311176969380dd3ffb5a794e30ee893320e357b03b43b199f',
    blockNumber: 9,
    from: '0xd183e4d372511781298698adf655dc403f374243',
    to: '0x2142ab79a9f2bd01d6535883aee5a7cac104b154',
    gasUsed: 19826,
    cumulativeGasUsed: 19826,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000',
    v: '0x26',
    r: '0x94a4f53eee10193c637ed4cd38d396358d063d2c6d15b684ba843be3c7c97f39',
    s: '0x49f6741e97957107bc63572ba696b5a11b5b9593bf827b3efd39f47d015ab1ef',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> 
```

And below is the output from Ganache. We can clearly see that the user with index 1, has now withdrawn his money which is 3 ETH.



CURRENT BLOCK 9
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE AUCTION

MNEMONIC
festival time visual habit habit fruit bacon cabbage future intact hazard wedding

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×dc72ff93E7F782E7DeF6710CF0a3ceD7414FEFea	104.99 ETH	5	0
0×d183e4D372511781298698aDf655DC403f374243	100.00 ETH	2	1
0×8BBEeeB9171F50dcAD4c3d59970cD6665531c16B	95.00 ETH	1	2
0×01b4B159432A4D1324CE97453f7f20BddeB005a5	100.00 ETH	1	3
0×0579A74Ebcb069fe896E0a5c11f361a10c944CE3	100.00 ETH	0	4
0×7e31300A52AA78261d96A1cd092EE80261C9a926	100.00 ETH	0	5
0×D37B6dBb62F96D99c8249E8c653D89080b168C1A	100.00 ETH	0	6
0×670106E9EFbDf08F188C44B05D21Ff28B86b2629	100.00 ETH	0	7
0×cE95C7614354aF12F76a134C3802BC5C9453f679	100.00 ETH	0	8
0×24219dEDa9e869f217f452914A6DEB68aA667ac3	100.00 ETH	0	9

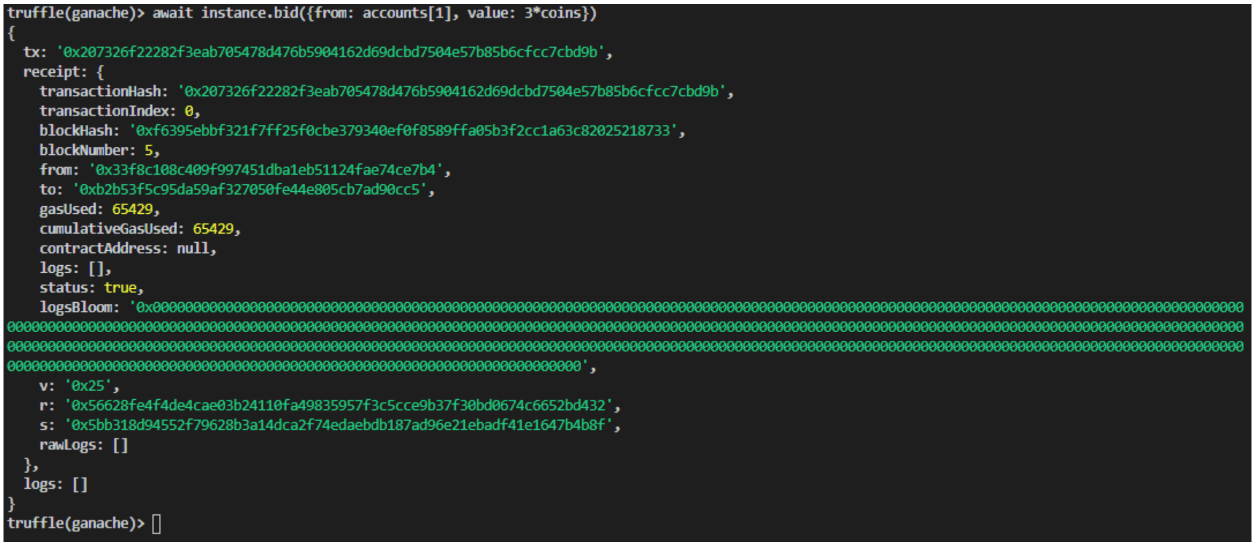
2. **Calculating Transaction Fee for Migrate Command and Above Functions:**
   a. **Migrate:** This command does 2 deployments and both times it uses gas. I'm calculating the transaction fee for both of them combined.

```
truffle(ganache)> migrate

Compiling your contracts...
===========================
> Compiling .\contracts\Auction.sol
> Compiling .\contracts\Migrations.sol
> Compilation warnings encountered:

    /C/Users/tummalanikhilesh/Downloads/hw1-source/contracts/Auction.sol:42:9: Warning: Failure condition of 'send' ignored. Consider using 'transfer' in
stead.
        beneficiary.send(highestBid);
        ^--------------------------^

> Artifacts written to C:\Users\tummalanikhilesh\Downloads\hw1-source\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang



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


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

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0x6419bb009d8206c8e758af6e9bf13b912342a9c5337408297a7978470a499ba1
   > Blocks: 0            Seconds: 0
   > contract address:    0xc95a7Ff62D4d1f8AA8D5cB0c25687B2A3DfbBBA7
   > block number:        1
   > block timestamp:     1582590927
   > account:             0x2d40b7cB60d618c93E555a10050F83E6818F97B8
   > balance:             99.9967165
   > gas used:            164175
   > 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
=====================

   Replacing 'Auction'
   -------------------
   > transaction hash:    0x59bf2519712aebfbf1ad0fe829a9c325d608c94bb7b2d15dfb9a7ce3880667de
   > Blocks: 0            Seconds: 0
   > contract address:    0x82b53f5C95DA59aF327050Fe44E805cB7ad90Cc5
   > block number:        3
   > block timestamp:     1582590927
   > account:             0x2d40b7cB60d618c93E555a10050F83E6818F97B8
   > balance:             99.98926984
   > gas used:            330032
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00660064 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00660064 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.00988414 ETH
```

Transaction Fee = gas used * gas price = (164175 + 330032) wei * 20gwei

= 494207wei * 20gwei = (9884140$*$ 10^9)wei

b. **bid():** Below is the output for bid() and it's transaction fee.

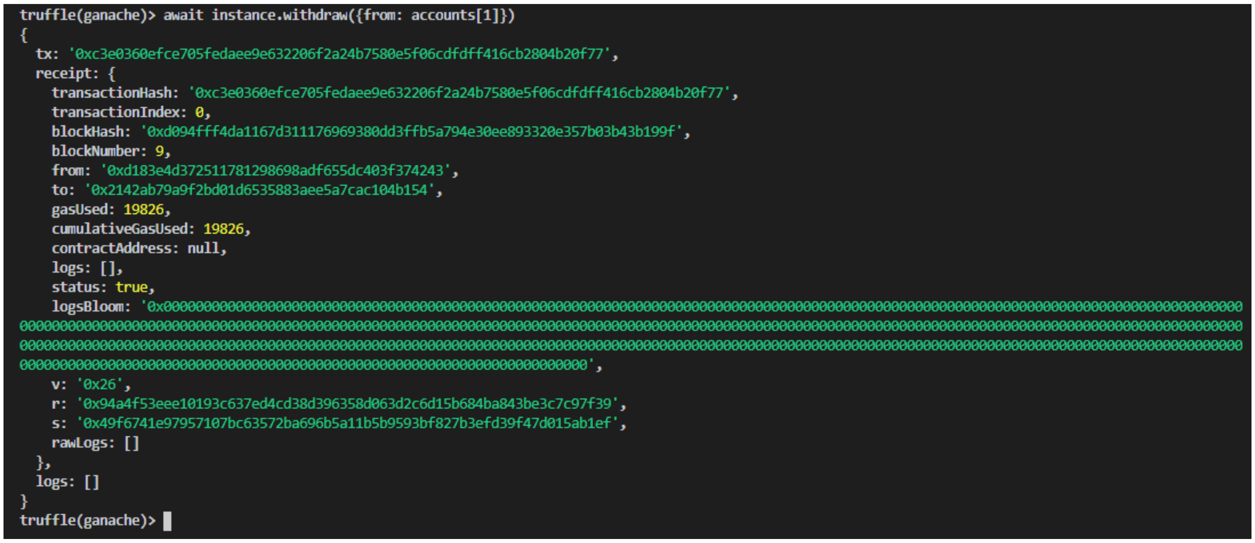


```
truffle(ganache)> await instance.bid({from: accounts[1], value: 3*coins})
{
  tx: '0x207326f22282f3eab705478d476b5904162d69dcbd7504e57b85b6cfcc7cbd9b',
  receipt: {
    transactionHash: '0x207326f22282f3eab705478d476b5904162d69dcbd7504e57b85b6cfcc7cbd9b',
    transactionIndex: 0,
    blockHash: '0xf6395ebbf321f7ff25f0cbe379340ef0f8589ffa05b3f2cc1a63c82025218733',
    blockNumber: 5,
    from: '0x33f8c108c409f997451dba1eb51124fae74ce7b4',
    to: '0xb2b53f5c95da59af327050fe44e805cb7ad90cc5',
    gasUsed: 65429,
    cumulativeGasUsed: 65429,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000',
    v: '0x25',
    r: '0x56628fe4f4de4cae03b24110fa49835957f3c5cce9b37f30bd0674c6652bd432',
    s: '0x5bb318d94552f79628b3a14dca2f74edaebdb187ad96e21ebadf41e1647b4b8f',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> 
```

Transaction Fee = gas used * gas price = 65429wei * 20gwei = (65429 * 20 * 10^9)wei

= (1308580 * 10^9)wei

c. **withdraw():**

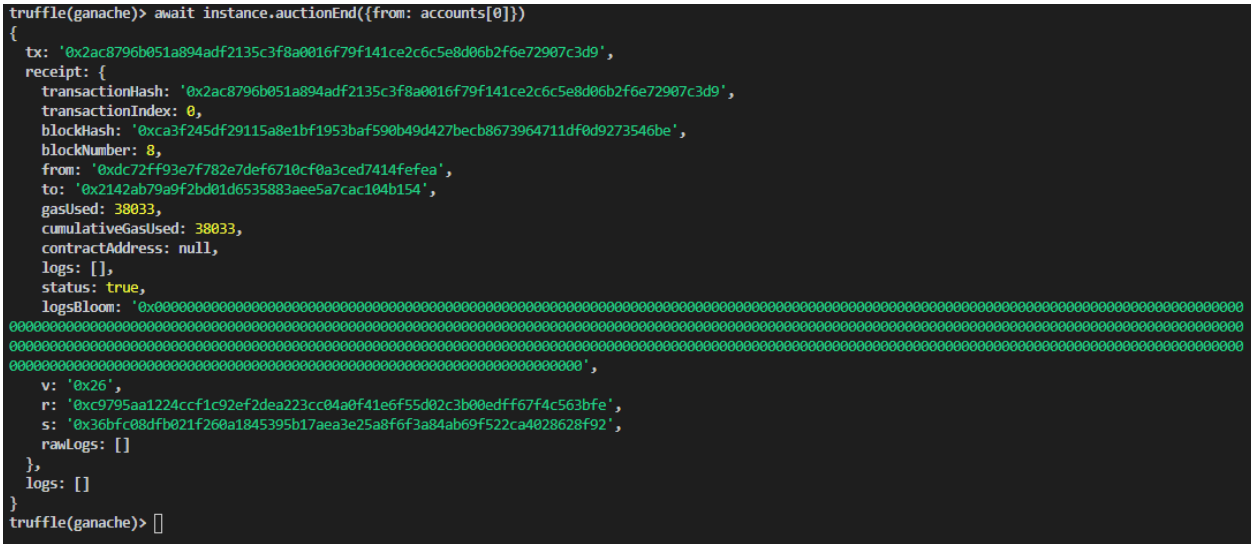


```
truffle(ganache)> await instance.withdraw({from: accounts[1]})
{
  tx: '0xc3e0360efce705fedaee9e632206f2a24b7580e5f06cdfdff416cb2804b20f77',
  receipt: {
    transactionHash: '0xc3e0360efce705fedaee9e632206f2a24b7580e5f06cdfdff416cb2804b20f77',
    transactionIndex: 0,
    blockHash: '0xd094fff4da1167d311176969380dd3ffb5a794e30ee893320e357b03b43b199f',
    blockNumber: 9,
    from: '0xd183e4d372511781298698adf655dc403f374243',
    to: '0x2142ab79a9f2bd01d6535883aee5a7cac104b154',
    gasUsed: 19826,
    cumulativeGasUsed: 19826,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000',
    v: '0x26',
    r: '0x94a4f53eee10193c637ed4cd38d396358d063d2c6d15b684ba843be3c7c97f39',
    s: '0x49f6741e97957107bc63572ba696b5a11b5b9593bf827b3efd39f47d015ab1ef',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> 
```

Transaction Fee = gas used * gas price = 19826wei * 20gwei = (19826 * 20 * 10^9)wei

= (396520 * 10^9)wei

d. auctionEnd():



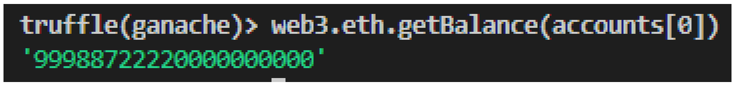
```
truffle(ganache)> await instance.auctionEnd({from: accounts[0]})
{
  tx: '0x2ac8796b051a894adf2135c3f8a0016f79f141ce2c6c5e8d06b2f6e72907c3d9',
  receipt: {
    transactionHash: '0x2ac8796b051a894adf2135c3f8a0016f79f141ce2c6c5e8d06b2f6e72907c3d9',
    transactionIndex: 0,
    blockHash: '0xca3f245df29115a8e1bf1953baf590b49d427becb8673964711df0d9273546be',
    blockNumber: 8,
    from: '0xdc72ff93e7f782e7def6710cf0a3ced7414fefea',
    to: '0x2142ab79a9f2bd01d6535883aee5a7cac104b154',
    gasUsed: 38033,
    cumulativeGasUsed: 38033,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000',
    v: '0x26',
    r: '0xc9795aa1224ccf1c92ef2dea223cc04a0f41e6f55d02c3b00edff67f4c563bfe',
    s: '0x36bfc08dfb021f260a1845395b17aea3e25a8f6f3a84ab69f522ca4028628f92',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> 
```

Transaction Fee = gas used * gas price = 38033wei * 20gwei = (38033 * 20 * 10^9)wei

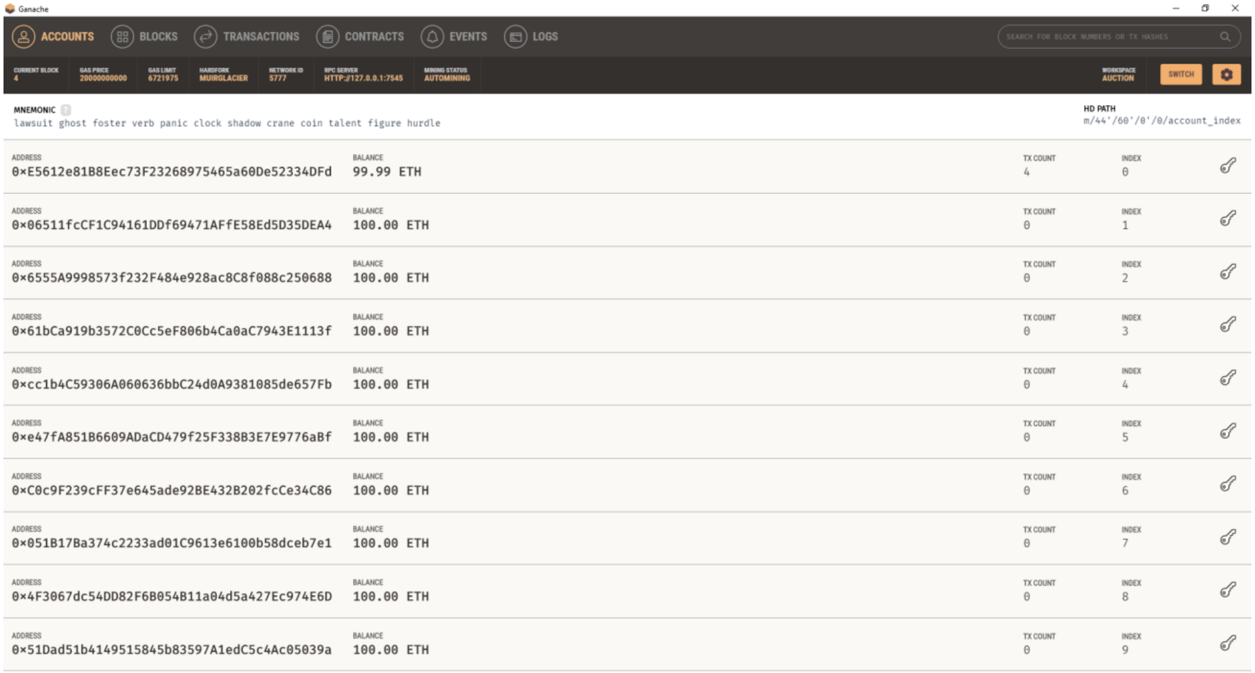
= (760660 * 10^9)wei

3. Difference in Beneficiary account Before and After auctionEnd():

Before the bidding starts and auction ends, the balance in the beneficiary account is shown below.



```
truffle(ganache)> web3.eth.getBalance(accounts[0])
'99988722220000000000'
```



CURRENT BLOCK 4
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE AUCTION

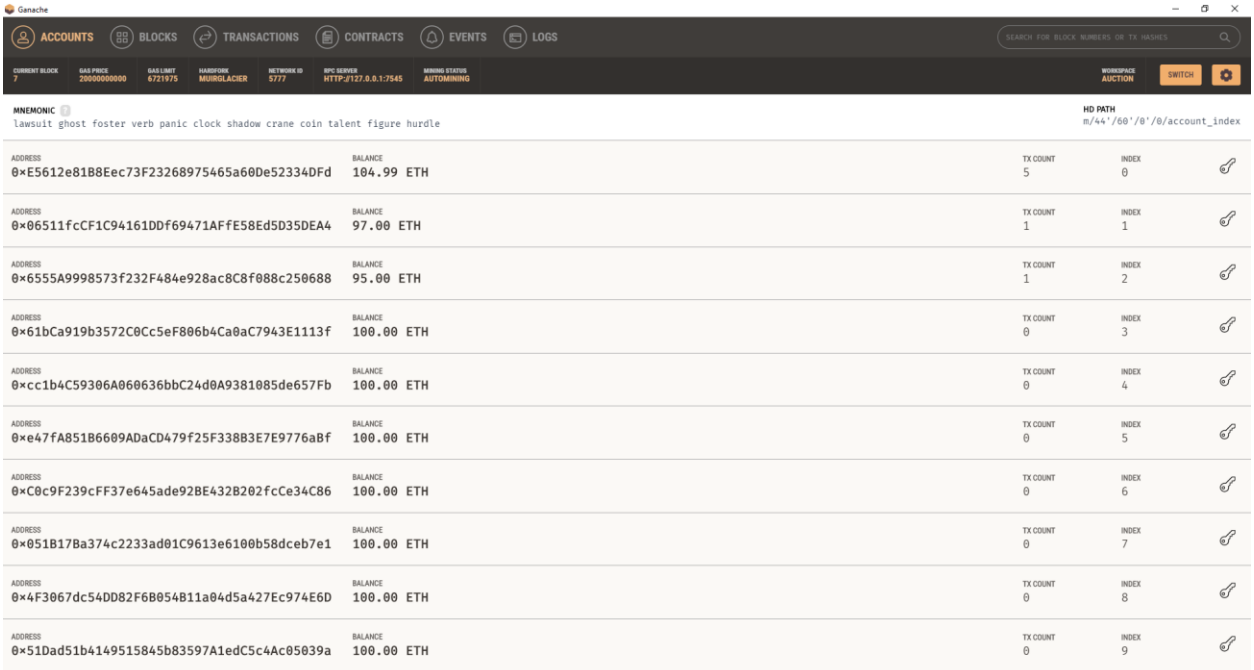
MNEMONIC
lawsuit ghost foster verb panic clock shadow crane coin talent figure hurdle

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×E5612e81B8Eec73F23268975465a60De52334DFd	99.99 ETH	4	0
0×06511fcCF1C94161DDf69471AFfE58Ed5D35DEA4	100.00 ETH	0	1
0×6555A9998573f232F484e928ac8C8f088c250688	100.00 ETH	0	2
0×61bCa919b3572C0Cc5eF806b4Ca0aC7943E1113f	100.00 ETH	0	3
0×cc1b4C59306A060636bbC24d0A9381085de657Fb	100.00 ETH	0	4
0×e47fA851B6609ADaCD479f25F338B3E7E9776aBf	100.00 ETH	0	5
0×C0c9F239cFF37e645ade92BE432B202fcCe34C86	100.00 ETH	0	6
0×051B17Ba374c2233ad01C9613e6100b58dceb7e1	100.00 ETH	0	7
0×4F3067dc54DD82F6B054B11a04d5a427Ec974E6D	100.00 ETH	0	8
0×51Dad51b4149515845b83597A1edC5c4Ac05039a	100.00 ETH	0	9

After the auctionEnd() function is executed, the balance in the beneficiary account is shown below:

```
truffle(ganache)> web3.eth.getBalance(accounts[0])
'104987961560000000000'
```



Calculation: The amount in the beneficiary account increase from 99.99 ETH to 104.99 ETH and the amount from the user with index 2, decreases from 100 ETH to 95 ETH. The calculation as to how it happens is shown below.

Balance[0] = Balance[0] before auctionEnd() + highest bid in the auction - gas used.

= 99988722220000000000 + 5*(Math.pow(10,18)) – 760660 * 10^9

= 104987961560000000000.

Here, we can see that the total balance we calculated is exactly same as the balance we obtained from the truffle console.