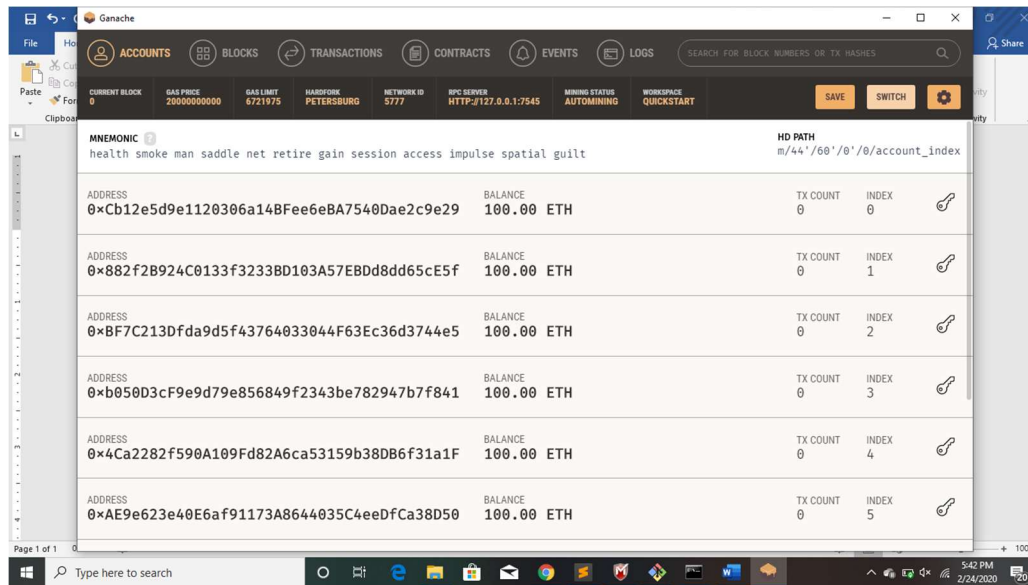


**Q.1) Brief explanation for each function you wrote and how it works. It would be great to show some experiments on sending/withdrawing bids to show that your contract works correctly**

- 1. function bid () payable ():** The **bid function** allows user to place the bid. Now whether the bid will be placed depends on the amount of new bid. If incoming bid is higher than the already existing bid then the new bid becomes the highest bid and old one is added to pending returns which shall be invoked by user to retrieve the money back in case their bid does not remain the highest. In case the incoming bid is smaller than highest bid the money is sent back to the bidder.
- 2. function withdraw () public returns (bool):** The function withdraw sends back the money to the user when they call withdraw function to get their money back in case, they lose the bid. Now, in order to handle the reentrancy, attack I have set a condition where before entering the logic of the code it checks if the pendingReturns for that particular id is 0 then it sends back an exception.
- 3. function auctionEnd () public:** Here, the beneficiary is allowed to call this function to end the auction. Once this function is invoked by the beneficiary the highest bid is transferred to the beneficiary and can be seen on the Ganache console.

## Initial states of the accounts



ADDRESS	BALANCE	TX COUNT	INDEX
0xCb12e5d9e1120306a14BFee6eBA7540Dae2c9e29	100.00 ETH	0	0
0x882f2B924C0133f323BD103A57EBDd8dd65cE5f	100.00 ETH	0	1
0xBF7C213Dfda9d5f43764033044F63Ec36d3744e5	100.00 ETH	0	2
0xb050D3cF9e9d79e856849f2343be782947b7f841	100.00 ETH	0	3
0x4Ca2282f590A109Fd82A6ca53159b38DB6f31a1F	100.00 ETH	0	4
0xAE9e623e40E6af91173A8644035C4eeDfCa38D50	100.00 ETH	0	5

## Console after compile and migrate

```
Compiling your contracts...
=====
> Compiling .\contracts\Auction.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to D:\UF sem 2\Blockchain\Homework1\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: 0x9e5f76c853a1206115ad2b01e6e5911a050d246b0bc94718195213ca97910335
> Blocks: 0
> contract address: 0x2CD187eEffa7bD03e6A6C04b12259F0Fd6373704
> block number: 1
> block timestamp: 1582582181
> account: 0xCb12e5d9e1120306a14BFee6eBA7540Dae2c9e29
> balance: 99.99623034
> gas used: 100103
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.00376966 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.00376966 ETH
```

## Gas used for deploying the contracts

```
C:\Windows\system32\cmd.exe - truffle console
> balance: 99.99623034
> gas used: 188483
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.00376966 ETH

> Saving migration to chain.
> Saving artifacts
> Total cost: 0.00376966 ETH

2_deploy_contracts.js
=====

Replacing 'Auction'
> transaction hash: 0xe9c22b17eb48d8ea8a451ebbd71420656aca83cdd6594373d947649234981e3c
> Blocks: 0
> contract address: 0x6a4402986469770E3AfA0fb950ea220e26d96147
> block number: 3
> block timestamp: 1582582102
> account: 0xCb12e5d9e1120306a14BFee6e8A7540Dae2c9e29
> balance: 99.9840842
> gas used: 494610
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.0098922 ETH

> Saving migration to chain.
> Saving artifacts
> Total cost: 0.0098922 ETH

Summary
=====
> Total deployments: 2
> Final cost: 0.01366186 ETH

truffle(ganache)>
```

Ganache

ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

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

SAVE SWITCH

TX HASH	FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE	
0x6808ba5cee86c8bc2a8aa297326a90bba2d0a39285de66e6231f2c264a04e5a	0xCb12e5d9e1120306a14BFee6e8A7540Dae2c9e29	0x2CD187eEffa7bD03e6A6C04b12259F0Fd6373704	27001	0	CONTRACT CALL
0xd2af2921ed16e118a85fa3abc6400b9d85a0d6ce8b3827bffcfdff772b4bf557	0xCb12e5d9e1120306a14BFee6e8A7540Dae2c9e29	0x6a4402986469770E3AfA0fb950ea220e26d96147	494610	0	CONTRACT CREATION
0xec1cc38f3d42c59ff420d8518f6cf59a23335844727a03ebfcbdb87767eea3f6	0xCb12e5d9e1120306a14BFee6e8A7540Dae2c9e29	0x2CD187eEffa7bD03e6A6C04b12259F0Fd6373704	42001	0	CONTRACT CALL
0x9e5f76c853a1206115ad2b01e6e5911a050d246b0bc94718195213ca97910335	0xCb12e5d9e1120306a14BFee6e8A7540Dae2c9e29	0x2CD187eEffa7bD03e6A6C04b12259F0Fd6373704	188483	0	CONTRACT CREATION

Summary

> Total dep

> Final cos

truffle(ganache)>

**Auction started by the beneficiary**

The screenshot displays the Ganache application window. The top navigation bar includes tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below this, a status bar shows various network parameters: CURRENT BLOCK (4), GAS PRICE (20000000000), GAS LIMIT (6721975), HARDFORK (PETERSBURG), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:7545), MINING STATUS (AUTOMINING), and WORKSPACE (QUICKSTART). The main area shows the MNEMONIC (health smoke man saddle net retire gain session access impulse spatial guilt) and the HD PATH (m/44'/60'/0'/0/account\_index). Below this, a table lists five accounts with their addresses, balances, transaction counts, and indices.

ADDRESS	BALANCE	TX COUNT	INDEX
0xCb12e5d9e1120306a14BFee6eBA7540Dae2c9e29	99.98 ETH	4	0
0x882f2B924C0133f3233BD103A57EBDd8dd65cE5f	100.00 ETH	0	1
0xBF7C213Dfda9d5f43764033044F63Ec36d3744e5	100.00 ETH	0	2
0xb050D3cF9e9d79e856849f2343be782947b7f841	100.00 ETH	0	3
0x4Ca2282f590A109Fd82A6ca53159b38DB6f31a1F	100.00 ETH	0	4
0xAE9e623e40E6af91173A8644035C4eeDfCa38D50	100.00 ETH	0	5









```

logs: []
}
truffle(ganache) . await instance.withdraw({from: acc[1]})
Thrown:
Error: Returned error: VM Exception while processing transaction: revert    at PromiEvent (C:\Users\choud\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\contract\lib\promievent.js:9:1)
    at TruffleContract.withdraw (C:\Users\choud\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\contract\lib\execute.js:169:1)
    at evalmachine.:1:18
    at evalmachine.:2:49
    at signHandlersWrap (vm.js:269:15)
    at Script.runInContext (vm.js:124:14)
    at runScript (C:\Users\choud\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\core\lib\console.js:222:1)
    at Console.interpret (C:\Users\choud\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\core\lib\console.js:237:1)
    at REPLManager.interpret (C:\Users\choud\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\core\lib\repl.js:131:1)
    at bound (domain.js:419:14)
    at REPLServer.runBound [as eval] (domain.js:432:12)
    at REPLServer.onLine (repl.js:715:10)
    at REPLServer.emit (events.js:218:5)
    at REPLServer.EventEmitter.emit (domain.js:475:20)
    at REPLServer.Interface._onLine (readline.js:316:10)
    at REPLServer.Interface._line (readline.js:693:8)
    at REPLServer.Interface._ttyWrite (readline.js:1019:14)
    at REPLServer.self._ttyWrite (repl.js:792:7)
    at ReadStream.onkeypress (readline.js:191:10)
    at ReadStream.emit (events.js:210:5)
    at ReadStream.EventEmitter.emit (domain.js:475:20)
    at emitKeys (internal/readline/utils.js:433:14) {
  hijackedStack: 'Error: Returned error: VM Exception while processing transaction: revert\n' +
    '    at Object.ErrorResponse (C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\web3-core-helpers\\src\\errors.js:29:1)'
}
\n' +
  '    at C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\web3-core-requestmanager\\src\\index.js:140:1\n' +
  '    at C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\packages\\provider\\wrapper.js:112:1\n' +
  '    at XMLHttpRequest.request.onreadystatechange (C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\web3-providers-http\\src\\index.js:96:1)\n' +
  '    at XMLHttpRequest.requestEventTarget.dispatchEvent (C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\xhr2-cookies\\dist\\xml-http-request-event-target.js:34:1)\n' +
  '    at XMLHttpRequest._setReadyState (C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\xhr2-cookies\\dist\\xml-http-request.js:208:1)\n' +
  '    at XMLHttpRequest._onHttpResponseEnd (C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\xhr2-cookies\\dist\\xml-http-request.js:310:1)\n' +
  '    at IncomingMessage. (C:\\Users\\choud\\AppData\\Roaming\\npm\\node_modules\\truffle\\build\\webpack:\\node_modules\\xhr2-cookies\\dist\\xml-http-request.js:289:47)\n' +
  '    at IncomingMessage.emit (events.js:215:7)\n' +
  '    at IncomingMessage.EventEmitter.emit (domain.js:498:23)\n' +

```

[illegible]



Beneficiary gets 6 ethers in his/her account as 6 was the highest bid placed. Value goes from 99.98 ethers to 105.98 ethers

The screenshot displays the Ganache application interface. The 'ACCOUNTS' tab is selected, showing a list of accounts. The first account, with address `0xCb12e5d9e1120306a148Fee6eBA7540Dae2c9e29`, has a balance of **105.98 ETH**, which is highlighted with a red box. The other accounts listed have balances of 100.00 ETH, 94.00 ETH, and 100.00 ETH. The interface includes a top navigation bar with tabs for Accounts, Blocks, Transactions, Contracts, Events, and Logs. A search bar is also present.

ADDRESS	BALANCE	TX COUNT	INDEX
<code>0xCb12e5d9e1120306a148Fee6eBA7540Dae2c9e29</code>	105.98 ETH	5	0
<code>0x882f2B924C0133f3233BD103A57EBDd8dd65cE5f</code>	100.00 ETH	2	1
<code>0xBF7C213Dfda9d5f43764033044F63Ec36d3744e5</code>	100.00 ETH	0	2
<code>0xb050D3cF9e9d79e856849f2343be782947b7f841</code>	94.00 ETH	1	3
<code>0x4Ca2282f590A109Fd82A6ca53159b38DB6f31a1F</code>	100.00 ETH	0	4
<code>0xAE9e623e40E6af91173A8644035C4eeDfCa38D50</code>	100.00 ETH	0	5

## 2. Transaction fees

### For migration

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

1_initial_migration.js
=====

  Replacing 'Migrations'
  -----
  > transaction hash:  0x9e5f76c853a1206115ad2b01e6e5911a050d246b0bc94718195213ca97910335
  > Blocks: 0         Seconds: 0
  > contract address: 0x2CD187eEffa7bD03e6A6C04b12259F0Fd6373704
  > block number:     1
  > block timestamp:   1582587801
  > account:          0xCb12e5d9e1120306a148Fee6eBA7540Dae2c9e29
  > balance:          99.99623034
  > gas used:         188483
  > gas price:         20 gwei
  > value sent:        0 ETH
  > total cost:        0.00376966 ETH

  > Saving migration to chain.
  > Saving artifacts
  -----
  > Total cost:        0.00376966 ETH
```

Transaction Fee: gas used \* gas price

$$= 188483 \text{wei} * 20 \text{gwei}$$

$$= 188483 * 20000000000$$

$$= 3769660000000000$$

### For function bid ()

[illegible]

**Transaction Fee: gas used \* gas price**

**=52733wei \* 20gwei**

**= 52733 \* 20000000000**

**=1054660000000000wei**



### For function auctionEnd ():

[illegible]

**Transaction Fee: gas used \* gas price**

**=20164wei \* 20gwei**

$$= 20164 * 20000000000$$

**=40328000000000wei**



## Q.3

Beneficiary/ account [0] balance before the auctionEnd() is called:

```
}  
truffle(ganache)> web3.eth.getBalance(acc[0])  
'99984958100000000000'
```

Same is in ganache 99.98ETH for account[0]

The screenshot shows the Ganache application window. The top navigation bar includes tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. The ACCOUNTS tab is active, displaying a table of accounts. The table has columns for ADDRESS, BALANCE, TX COUNT, and INDEX. The first account (index 0) has a balance of 99.98 ETH. The other accounts (indices 1-5) have a balance of 100.00 ETH. The interface also shows a mnemonic phrase and an HD path.

ADDRESS	BALANCE	TX COUNT	INDEX
0xCb12e5d9e1120306a148Fee6eBA7540Dae2c9e29	99.98 ETH	4	0
0x882f2B924C0133f3233BD103A57EBDd8dd65cE5f	100.00 ETH	0	1
0xBF7C213Dfda9d5f43764033044F63Ec36d3744e5	100.00 ETH	0	2
0xb050D3cF9e9d79e856849f2343be782947b7f841	100.00 ETH	0	3
0x4Ca2282f590A109Fd82A6ca53159b38DB6f31a1F	100.00 ETH	0	4
0xAE9e623e40E6af91173A8644035C4eeDfCa38D50	100.00 ETH	0	5

**Beneficiary/ Account [0] after auctionEnd() is called. Value goes from 99.98 to 105.98 ( total increase of 6 ethers that matches the highest bid )**

```
truffle(ganache)> web3.eth.getBalance(acc[0])
'10598455482000000000'
truffle(ganache)>
```

ADDRESS	BALANCE	TX COUNT	INDEX
0xCb12e5d9e1120306a148Fee6eBA7540Dae2c9e29	105.98 ETH	5	0
0x882f2B924C0133f3233BD103A57EBDd8dd65cE5f	100.00 ETH	2	1
0xBF7C213Dfda9d5f43764033044F63Ec36d3744e5	100.00 ETH	0	2
0xb050D3cF9e9d79e856849f2343be782947b7f841	94.00 ETH	1	3
0x4Ca2282f590A109Fd82A6ca53159b38DB6f31a1F	100.00 ETH	0	4
0xAE9e623e40E6af91173A8644035C4eeDfCa38D50	100.00 ETH	0	5

**Calculation:**

**Function auctionEnd():**

**Balance[0]=Balance[0] (before auctionEnd) + highest bid in auction – gas fees used**

**=9998495810000000000 + 6\* Math.pow(10,18) – 29356**

**=10598455482000000000**

**Thus, our calculation matches the displayed figure after auction ends.**