The Smart Contract maintains a couple of variables. Below is the information for the same.

```
    //holds the address of all the registered auctioneers
    address[] auctioneer;

    //boolean variable to hold of registration process is on
    bool isRegistrationStopped;

    // boolean variable to know if bidding is on or of
    bool bidStopped;

    // holds the map of address and the commited value
    mapping(address => bytes32)
```

```
public commitment;

    //keys for commitment map
    address[] private
commitmentKeys;

    //map of address and bid in usd
    mapping(address => uint)
public bidMap;
    address[] private bidMapKeys;

    // list if all valid auctioneers
    address payable[]
validAuctioneer;

    //owner of the smart contract
    address owner;
    //map to keep note of the money
locked which needs to be returned
```

```
    mapping(address => uint)
private auctioneerLockedEther;
    address[] private
auctioneerLockedEtherKeys;

    //list of auction_winners
    mapping(uint => address)
auction_winners;
    uint auction_number;
    // exchange rate
    uint public ETHUSD;
```

Additional useful functions

getWinners: get list of winners

getBalance: get balance of smart contract

Test Case Information below

5 users (User 5 and user 4 are inValid users)

User 4 places less value in commitment and user 5 hash doesn’t match case

1st User
Address: 0xdA5EC4f3e814EDD335Fe7154C11BC2Bd4e622B97
Bid: 190 USD
Random_seed: 1
Hash for 1st: 0xe9e5335a6df74f7c756d3a5dc5b9613678cac204064bc8117bc641392424615a
Commitment: 5 ethers

2nd User
Address:
0xF8b81688544617FEdfC9476e1
24a94Ef3669Da62
Bid: 380 USD
Random_seed: 2
Hash for 2nd:
0x21dc236545efd2fc4351378dff99f054fc3
71432ee335aa3d3f93d1d9b0d0d34
Commitment: 5 ethers

3rd User - 5
Address:
0x2F964B3815269eA062979b22C
75534A4B2eb12Ab
Bid: 570 USD
Random_seed: 3
Hash for 1st:
0x941ee0bd7e9c431dd1eab2acbce65c

9d4d7baad0af9fd527dfba10c5e82a1c5d

Commitment: 5 ethers

Less Commitment case

4th User - 2
Address: 0x0Ede2B276B7db4f6D572908FF5daAb3270A30dc2
Bid: 760 USD
Random_seed: 4
Hash: 0x24ad6262452c5d2fed5d01d15e9bf77414c7c97d131aab1a802d1b6a3c3db22a

Commitment: 2 ethers

Hash Mismatch Case

5th User - 15
Address:
0xfbaA5E377f5E36c41b2fF15ED
D725C45750Ef7e5
Bid: 1900
Random_seed: 5
Hash:
0x5faa094ed6f0a1b4e26191a332d
c5b71512e61353ebb19bae953aed
5c828057f

Expected Results

Winner - User 3 (Address:

0x2F964B3815269eA062979b22C75534A4B2eb12Ab)
Smart Contract Balance After finish: 20 ethers

Logs below:

transact to Auction.computeWinner pending ...
[vm]
from:0x054...cbdd1
to: Auction.computeWinner() 0x167...0924d
value:0 wei
data:0x9d9...d3b2d
logs:0
hash:0x528...e7446

Debug

status	0x1 Transaction mined and execution succeed
transaction hash	0x528ede2ff73b03e4e8a0509e00d6a5af34cb01c75b7e9eff646178abc5be7446
from	0x05409d15db2d368b922bb00547ad1706df4cbdd1
to	Auction.computeWinner() 0x1679591c985bde2415c7c8faa2cd82798680924d
gas	30000000000 gas
transaction cost	115319 gas
execution cost	94047 gas
hash	0x528ede2ff73b03e4e8a0509e00d6a5af34cb01c75b7e9eff646178abc5be7446

input	0x9d9...d3b2d
decoded input	{}
decoded output	**{ "0": "address: 0x2F964B3815269eA062979b22C75534A4B2eb12Ab" }**
logs	[]
value	0 wei

transact to Auction.getBalance pending ...
[vm]
from:0x054...cbdd1
to: Auction.getBalance() 0x167...0924d
value:0 wei
data:0x120...65fe0
logs:0
hash:0x8bb...9e47b

Debug

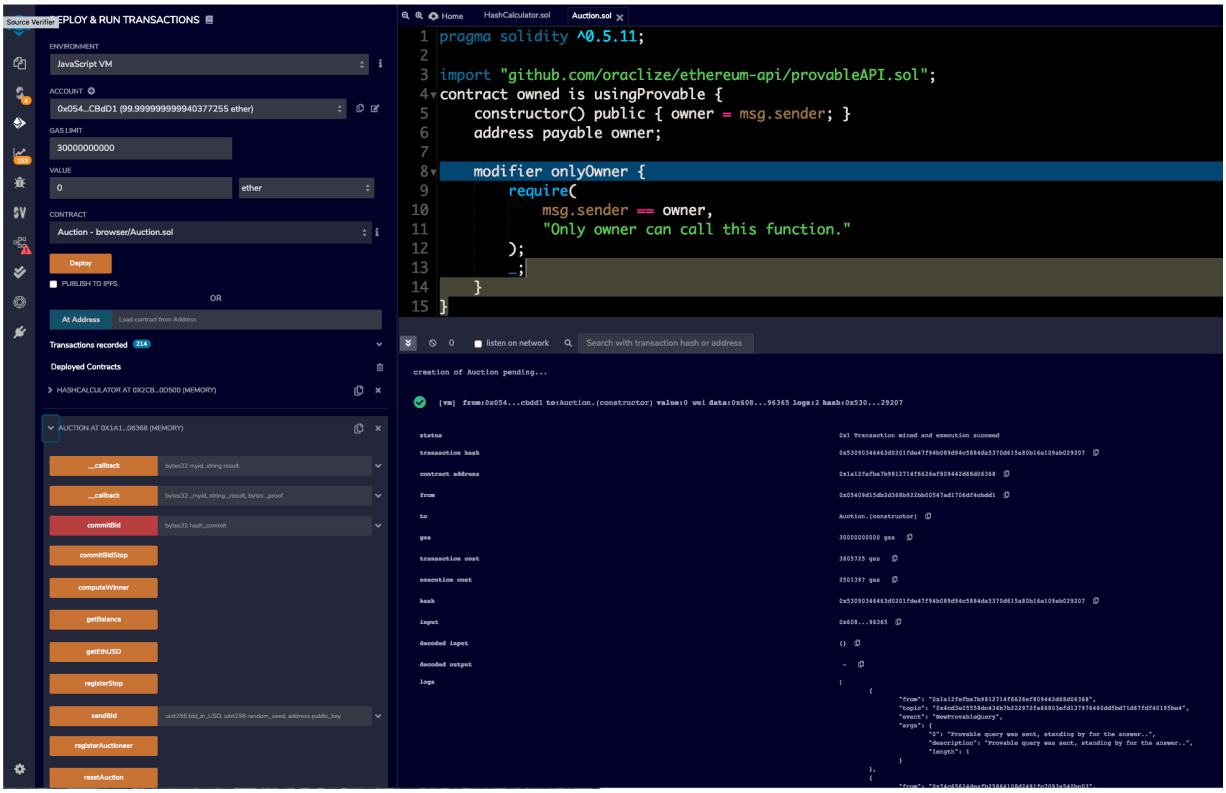
status	0x1 Transaction mined and execution succeed
transaction hash	0x8bbd8ab1706cf576f0f16f8883c59d59f5d6e19866998820ff40aee48399e47b
from	0x05409d15db2d368b922bb00547ad1706df4cbdd1

to	Auction.getBalance() 0x1679591c985bde2415c7c8faa2cd82798680924d
gas	30000000000 gas
transaction cost	21511 gas
execution cost	239 gas
hash	0x8bbd8ab1706cf576f0f16f8883c59d59f5d6e19866998820f f40aee48399e47b
input	0x120...65fe0
decoded input	{}
decoded output	**{ "0": "uint256: 20031914893617021276" }**
logs	[]
value	0 wei

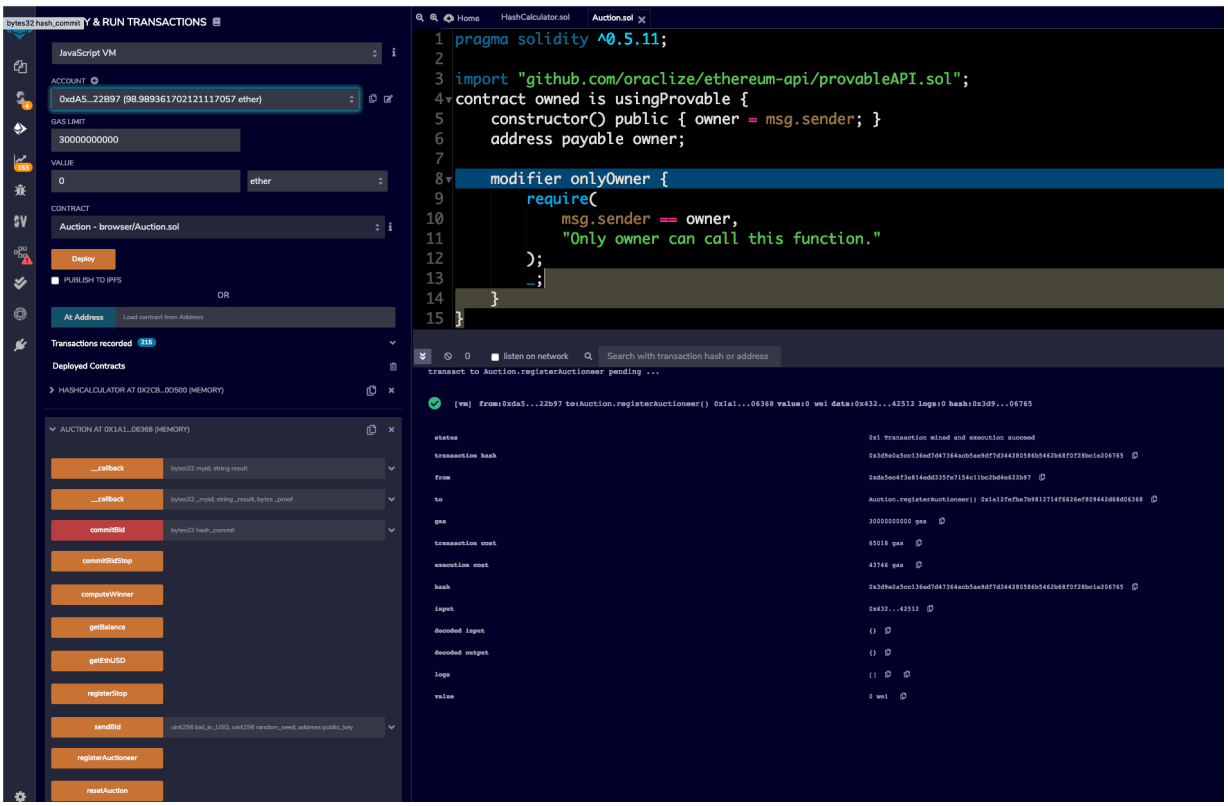
>

Below are the screenshots of various scenarios

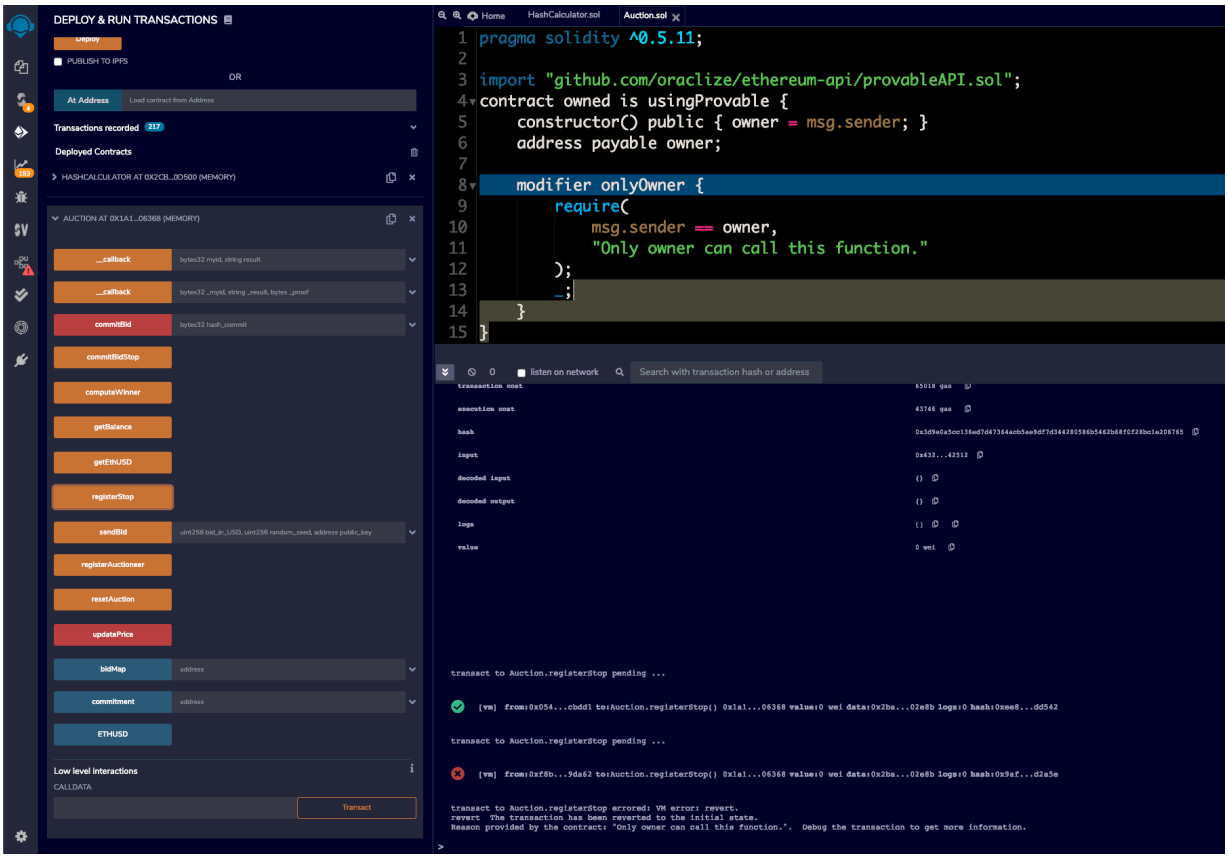
#1 Deployment



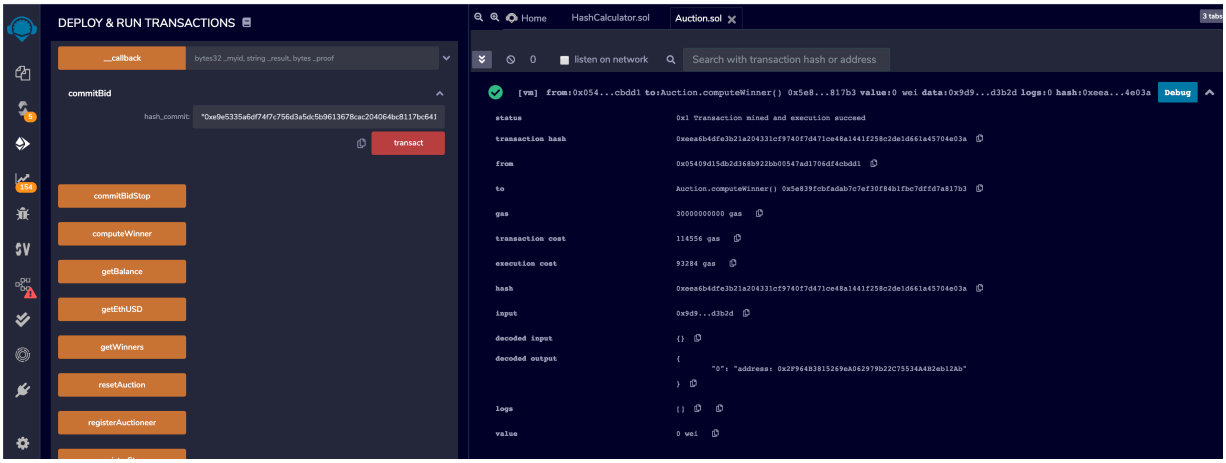
#2 registration of users



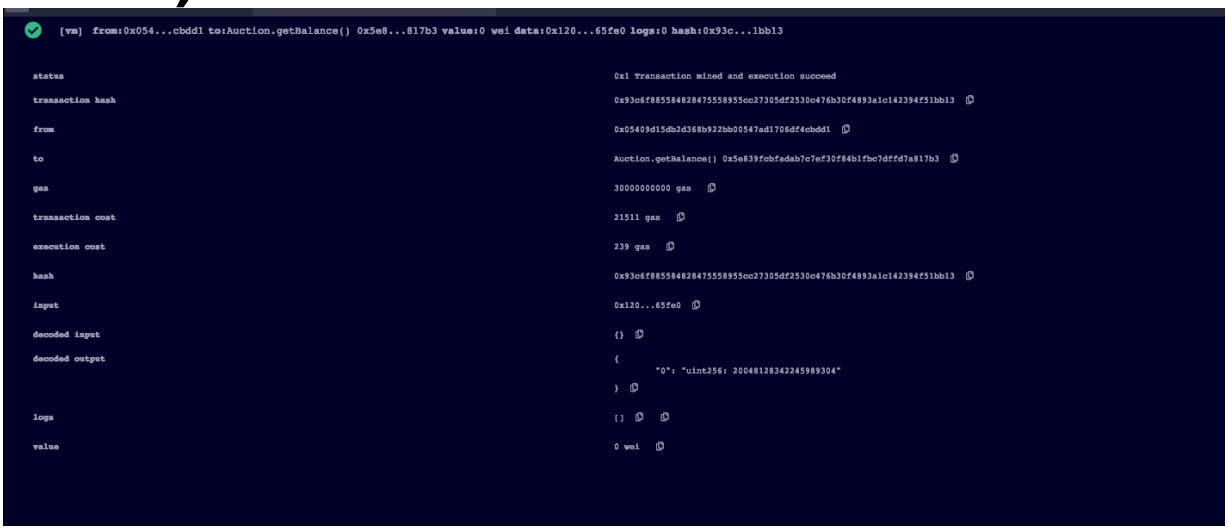
#3 Registration Error after its being stopped



#4 Compute Winner in above Test Case Scenario with 5 users 3rd User Wins the Auction



Smart Contract Balance after Auction (value of auction winnerr (3 ether) + value sent by 4th user (2 ether) + value sent by 5th user (15 ether) = 20 ether



getWinners() output for a use case