IBC Project

Group Members:

- i16-0188
- i16-0418

Project title: Online Bidding System

Project description:

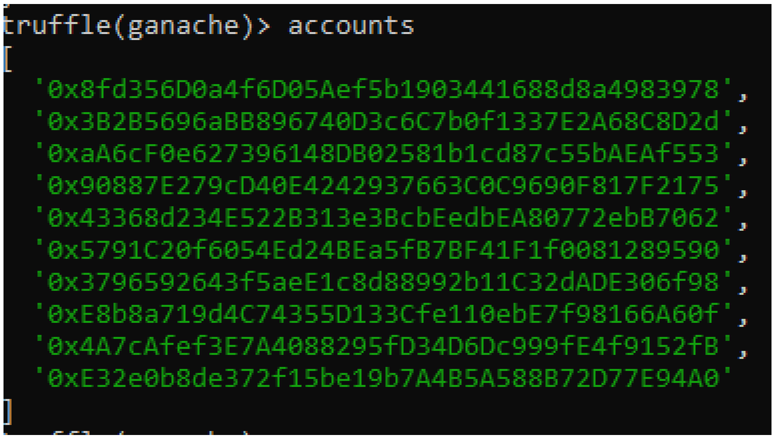
User will be able to add his products and add description about those products. Once the product is added it will be available for bidding only if the current owner of the product has set the initial bid amount. After setting the initial bid amount anyone can add his bid for that product which must be greater than the initial bid amount set by the owner.

The owner of the product can then accept any bid and that product will become the property of the accepted bidder in exchange of the bid amount.

Requirements:

1. Add products
2. Set Product for Bidding
3. Add bids (Anyone can add bids)
4. Accept bid (By product owner)

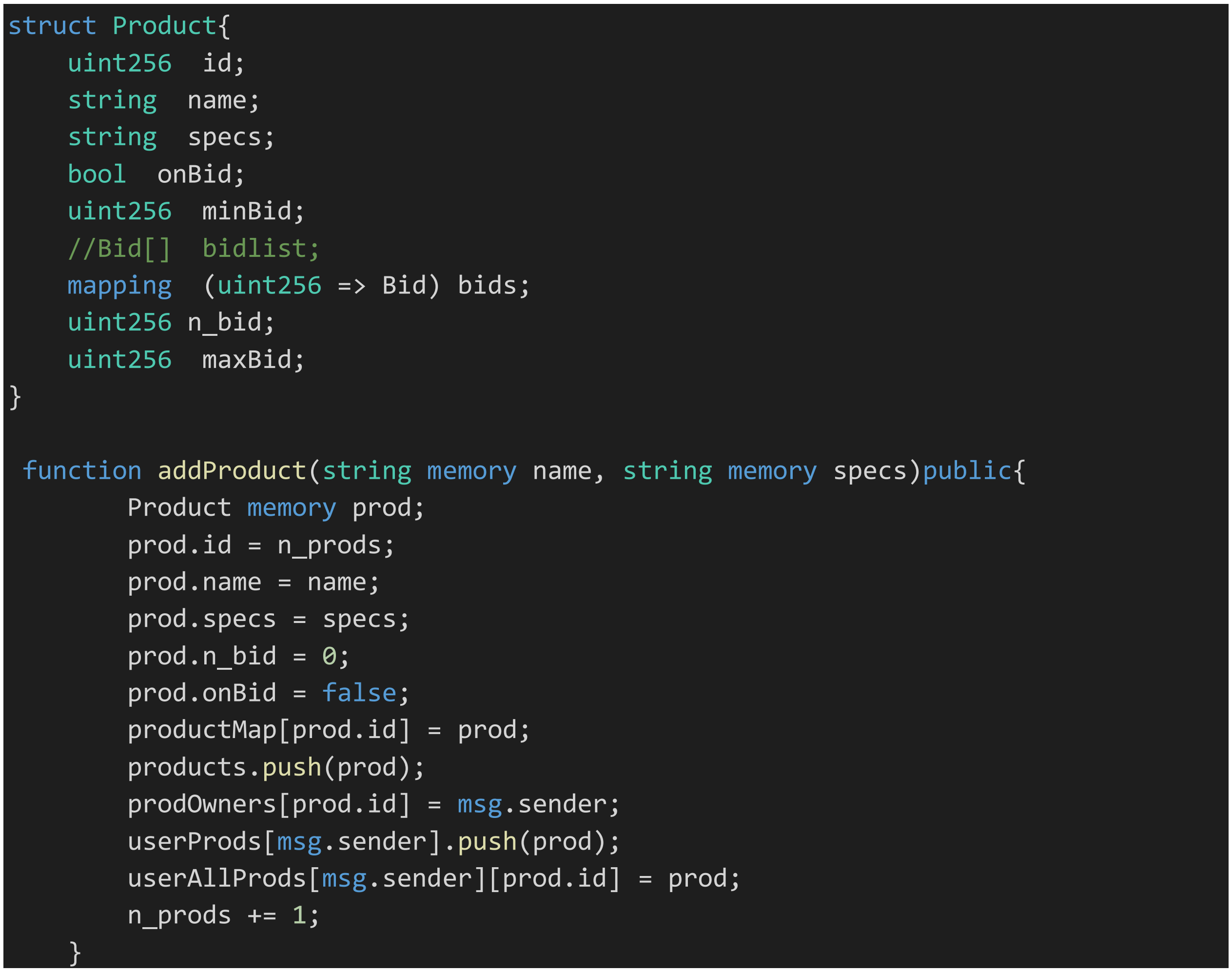
Current Ganache Accounts:



```
truffle(ganache)> accounts
[
  '0x8fd356D0a4f6D05Aef5b1903441688d8a4983978',
  '0x3B2B5696aBB896740D3c6C7b0f1337E2A68C8D2d',
  '0xaA6cF0e627396148DB02581b1cd87c55bAEAf553',
  '0x90887E279cD40E4242937663C0C9690F817F2175',
  '0x43368d234E522B313e3BcbEedbEA80772ebB7062',
  '0x5791C20f6054Ed24BEa5fB7BF41F1f0081289590',
  '0x3796592643f5aeE1c8d88992b11C32dADE306f98',
  '0xE8b8a719d4C74355D133Cfe110ebE7f98166A60f',
  '0x4A7cAfef3E7A4088295fD34D6Dc999fE4f9152fB',
  '0xE32e0b8de372f15be19b7A4B5A588B72D77E94A0'
]
```

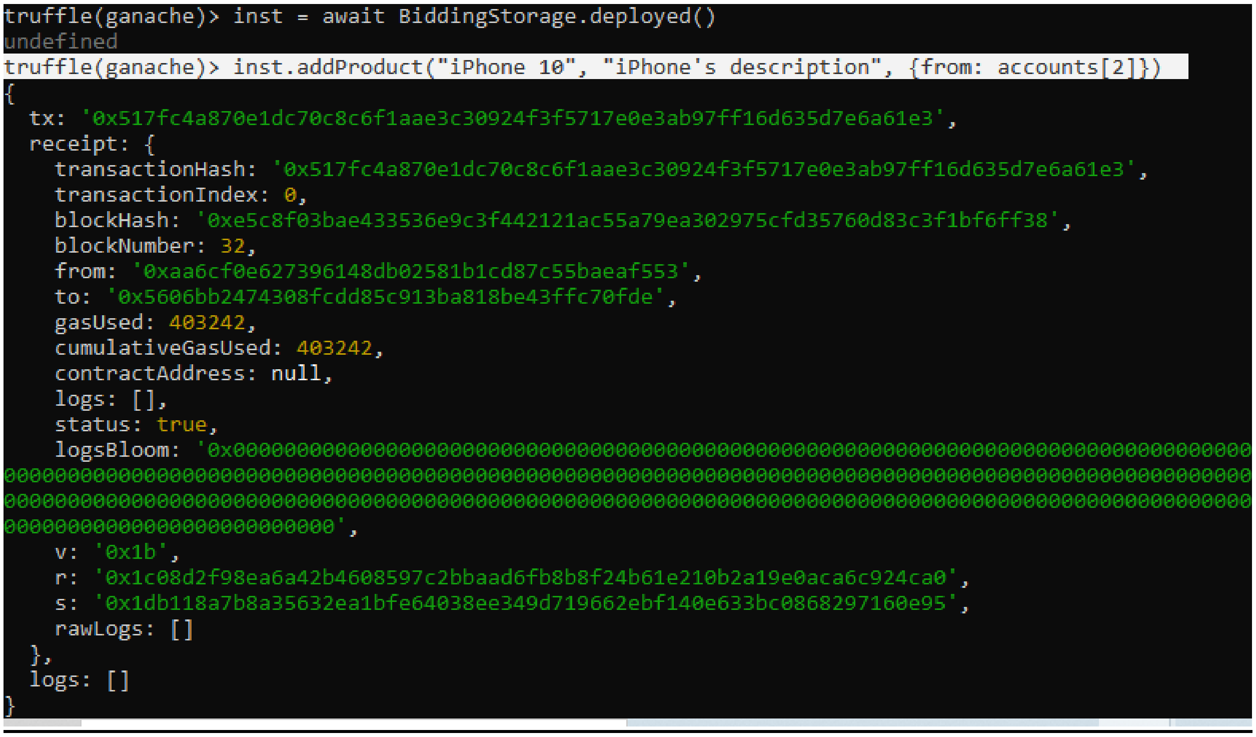
1. **Add Products**

   Solidity Code:



```
struct Product{
    uint256  id;
    string  name;
    string  specs;
    bool  onBid;
    uint256  minBid;
    //Bid[]  bidlist;
    mapping  (uint256 => Bid) bids;
    uint256 n_bid;
    uint256  maxBid;
}

 function addProduct(string memory name, string memory specs)public{
        Product memory prod;
        prod.id = n_prods;
        prod.name = name;
        prod.specs = specs;
        prod.n_bid = 0;
        prod.onBid = false;
        productMap[prod.id] = prod;
        products.push(prod);
        prodOwners[prod.id] = msg.sender;
        userProds[msg.sender].push(prod);
        userAllProds[msg.sender][prod.id] = prod;
        n_prods += 1;
    }
```

Ganache:

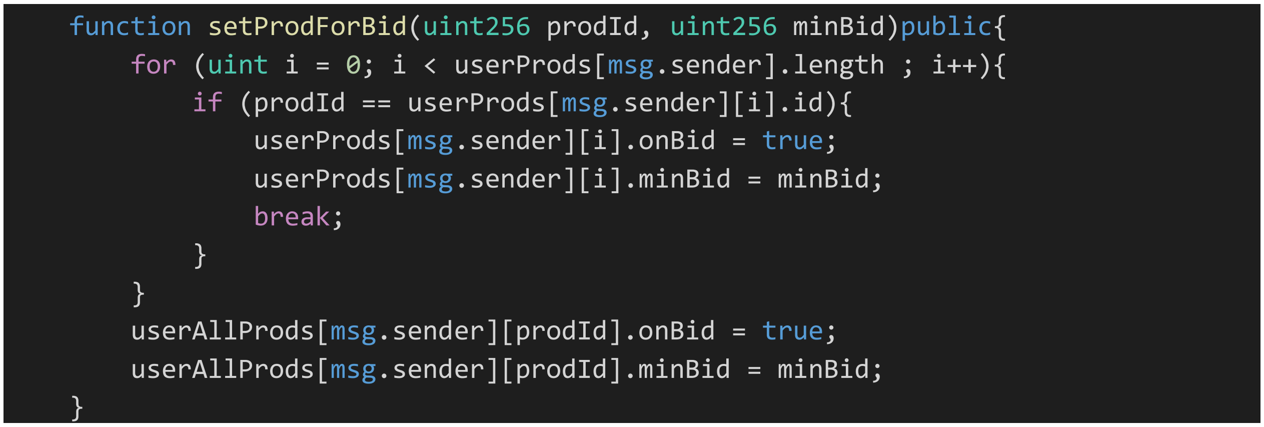


```
truffle(ganache)> inst = await BiddingStorage.deployed()
undefined
truffle(ganache)> inst.addProduct("iPhone 10", "iPhone's description", {from: accounts[2]})
{
  tx: '0x517fc4a870e1dc70c8c6f1aae3c30924f3f5717e0e3ab97ff16d635d7e6a61e3',
  receipt: {
    transactionHash: '0x517fc4a870e1dc70c8c6f1aae3c30924f3f5717e0e3ab97ff16d635d7e6a61e3',
    transactionIndex: 0,
    blockHash: '0xe5c8f03bae433536e9c3f442121ac55a79ea302975cfd35760d83c3f1bf6ff38',
    blockNumber: 32,
    from: '0xaa6cf0e627396148db02581b1cd87c55baeaf553',
    to: '0x5606bb2474308fcdd85c913ba818be43ffc70fde',
    gasUsed: 403242,
    cumulativeGasUsed: 403242,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000',
    v: '0x1b',
    r: '0x1c08d2f98ea6a42b4608597c2bbaad6fb8b8f24b61e210b2a19e0aca6c924ca0',
    s: '0x1db118a7b8a35632ea1bfe64038ee349d719662ebf140e633bc0868297160e95',
    rawLogs: []
  },
  logs: []
}
```

So when addProduct() function is called from account 2 with product's name as "IPhone 10" and product's description as "iPhone's description" , the product is added with Product's ID as '3' and now is in the ownership of account 2.

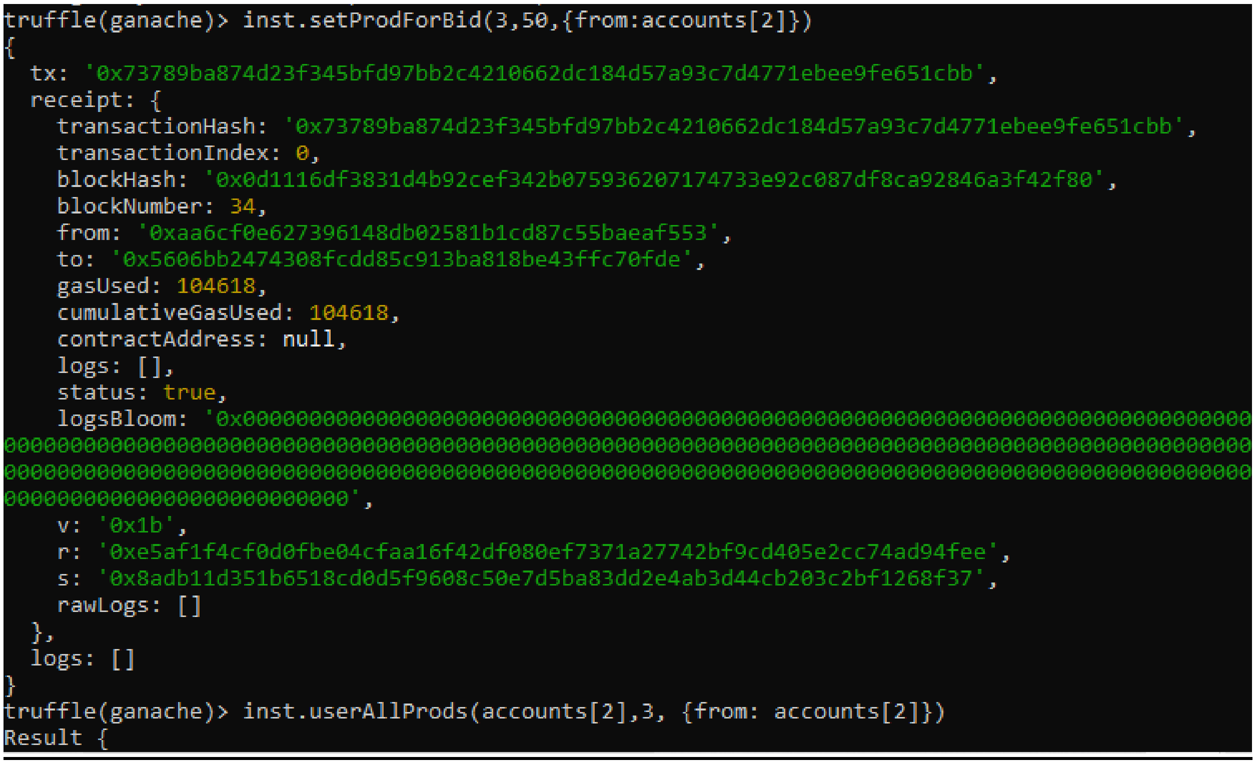
2. Set product for bidding

Solidity Code:

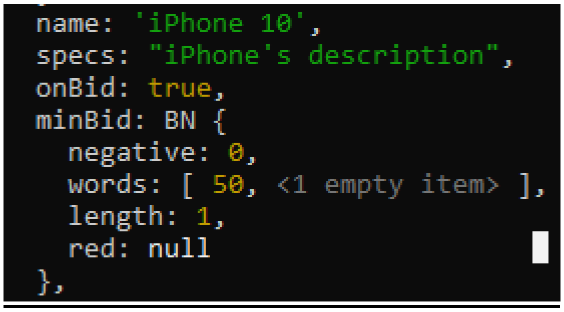


```
function setProdForBid(uint256 prodId, uint256 minBid)public{
    for (uint i = 0; i < userProds[msg.sender].length ; i++){
        if (prodId == userProds[msg.sender][i].id){
            userProds[msg.sender][i].onBid = true;
            userProds[msg.sender][i].minBid = minBid;
            break;
        }
    }
    userAllProds[msg.sender][prodId].onBid = true;
    userAllProds[msg.sender][prodId].minBid = minBid;
}
```

<u>Ganache:</u>



```
truffle(ganache)> inst.setProdForBid(3,50,{from:accounts[2]})
{
  tx: '0x73789ba874d23f345bfd97bb2c4210662dc184d57a93c7d4771ebee9fe651cbb',
  receipt: {
    transactionHash: '0x73789ba874d23f345bfd97bb2c4210662dc184d57a93c7d4771ebee9fe651cbb',
    transactionIndex: 0,
    blockHash: '0x0d1116df3831d4b92cef342b07593620717473 3e92c087df8ca92846a3f42f80',
    blockNumber: 34,
    from: '0xaa6cf0e627396148db02581b1cd87c55baeaf553',
    to: '0x5606bb2474308fcdd85c913ba818be43ffc70fde',
    gasUsed: 104618,
    cumulativeGasUsed: 104618,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000',
    v: '0x1b',
    r: '0xe5af1f4cf0d0fbe04cfaa16f42df080ef7371a27742bf9cd405e2cc74ad94fee',
    s: '0x8adb11d351b6518cd0d5f9608c50e7d5ba83dd2e4ab3d44cb203c2bf1268f37',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> inst.userAllProds(accounts[2],3, {from: accounts[2]})
Result {
```

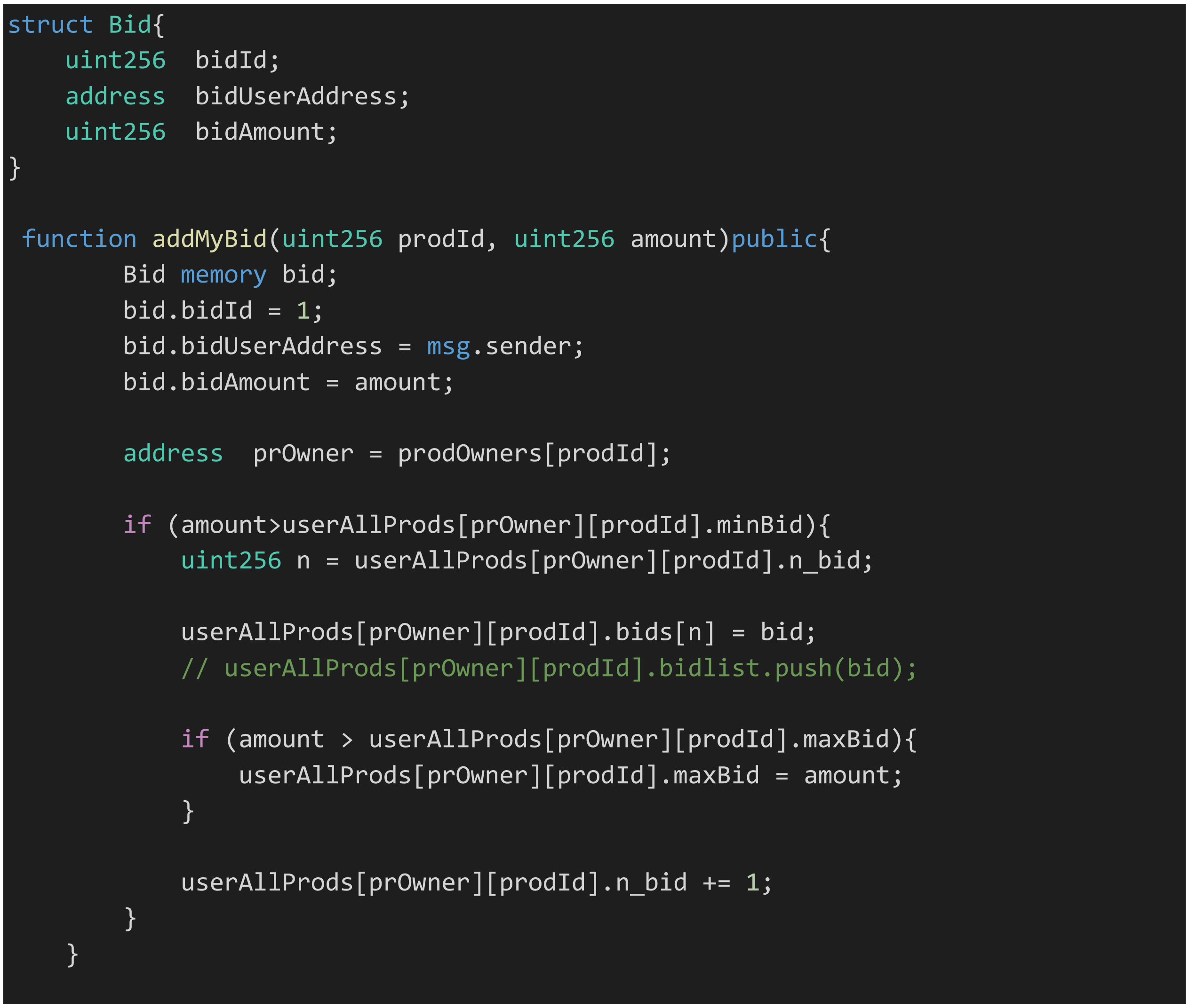


```
  name: 'iPhone 10',
  specs: "iPhone's description",
  onBid: true,
  minBid: BN {
    negative: 0,
    words: [ 50, <1 empty item> ],
    length: 1,
    red: null
  },
```

Now when setBidForProd() is called with initial minimum bid amount passed as “50”, the bidding is enabled (onBid is set to ‘true’) and minimum bid amount is set to ‘50’ for iPhone. Now any other account can bid for this product with amount greater than 50.

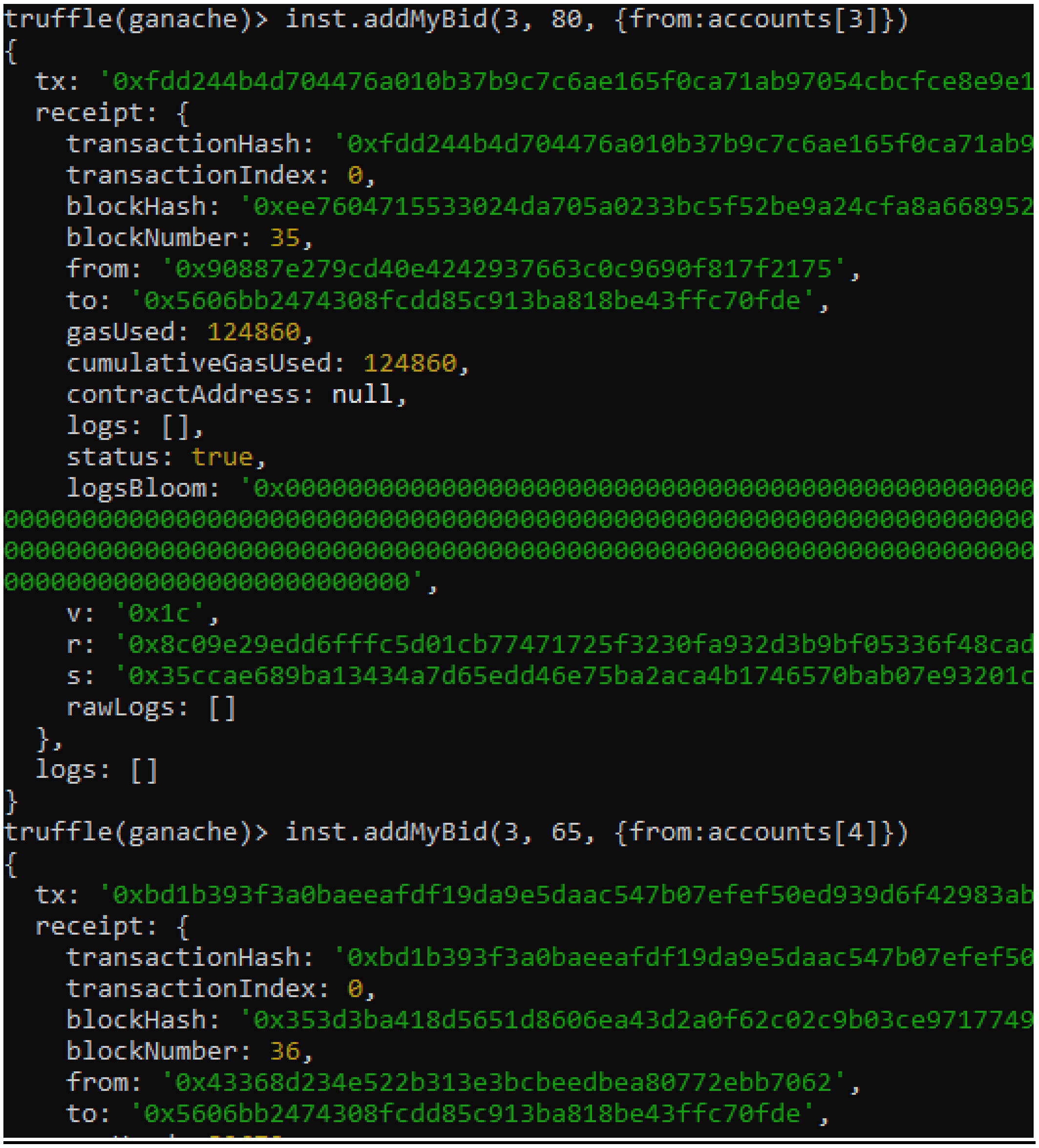
3. Add bids (Anyone can add bids)

Solidity Code:



```
struct Bid{
    uint256  bidId;
    address  bidUserAddress;
    uint256  bidAmount;
}

 function addMyBid(uint256 prodId, uint256 amount)public{
        Bid memory bid;
        bid.bidId = 1;
        bid.bidUserAddress = msg.sender;
        bid.bidAmount = amount;

        address  prOwner = prodOwners[prodId];

        if (amount>userAllProds[prOwner][prodId].minBid){
            uint256 n = userAllProds[prOwner][prodId].n_bid;

            userAllProds[prOwner][prodId].bids[n] = bid;
            // userAllProds[prOwner][prodId].bidlist.push(bid);

            if (amount > userAllProds[prOwner][prodId].maxBid){
                userAllProds[prOwner][prodId].maxBid = amount;
            }

            userAllProds[prOwner][prodId].n_bid += 1;
        }
    }
```

Ganache:

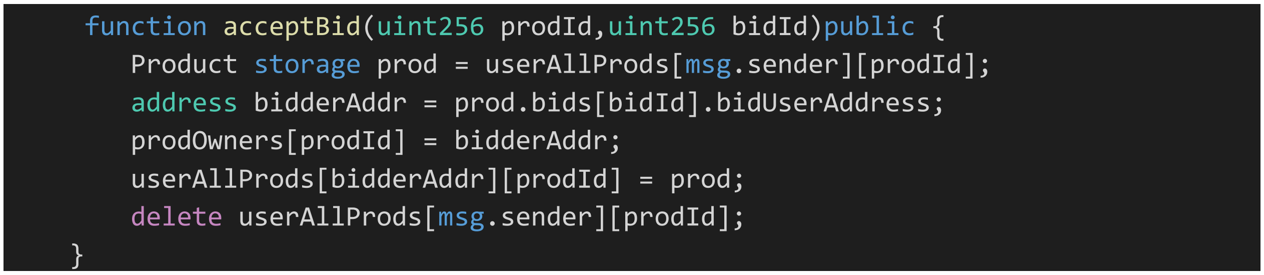


```
truffle(ganache)> inst.addMyBid(3, 80, {from:accounts[3]})
{
  tx: '0xfdd244b4d704476a010b37b9c7c6ae165f0ca71ab97054cbcfce8e9e1
  receipt: {
    transactionHash: '0xfdd244b4d704476a010b37b9c7c6ae165f0ca71ab9
    transactionIndex: 0,
    blockHash: '0xee7604715533024da705a0233bc5f52be9a24cfa8a668952
    blockNumber: 35,
    from: '0x90887e279cd40e4242937663c0c9690f817f2175',
    to: '0x5606bb2474308fcdd85c913ba818be43ffc70fde',
    gasUsed: 124860,
    cumulativeGasUsed: 124860,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000',
    v: '0x1c',
    r: '0x8c09e29edd6fffc5d01cb77471725f3230fa932d3b9bf05336f48cad
    s: '0x35ccae689ba13434a7d65edd46e75ba2aca4b1746570bab07e93201c
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> inst.addMyBid(3, 65, {from:accounts[4]})
{
  tx: '0xbd1b393f3a0baeeafdf19da9e5daac547b07efef50ed939d6f42983ab
  receipt: {
    transactionHash: '0xbd1b393f3a0baeeafdf19da9e5daac547b07efef50
    transactionIndex: 0,
    blockHash: '0x353d3ba418d5651d8606ea43d2a0f62c02c9b03ce9717749
    blockNumber: 36,
    from: '0x43368d234e522b313e3bcbeedbea80772ebb7062',
    to: '0x5606bb2474308fcdd85c913ba818be43ffc70fde',
```

Now account 3 and account 4 has called addMyBid() function and added their bids of amount 80 and 65 respectively, to the product (iPhone with id '3') of account 2.

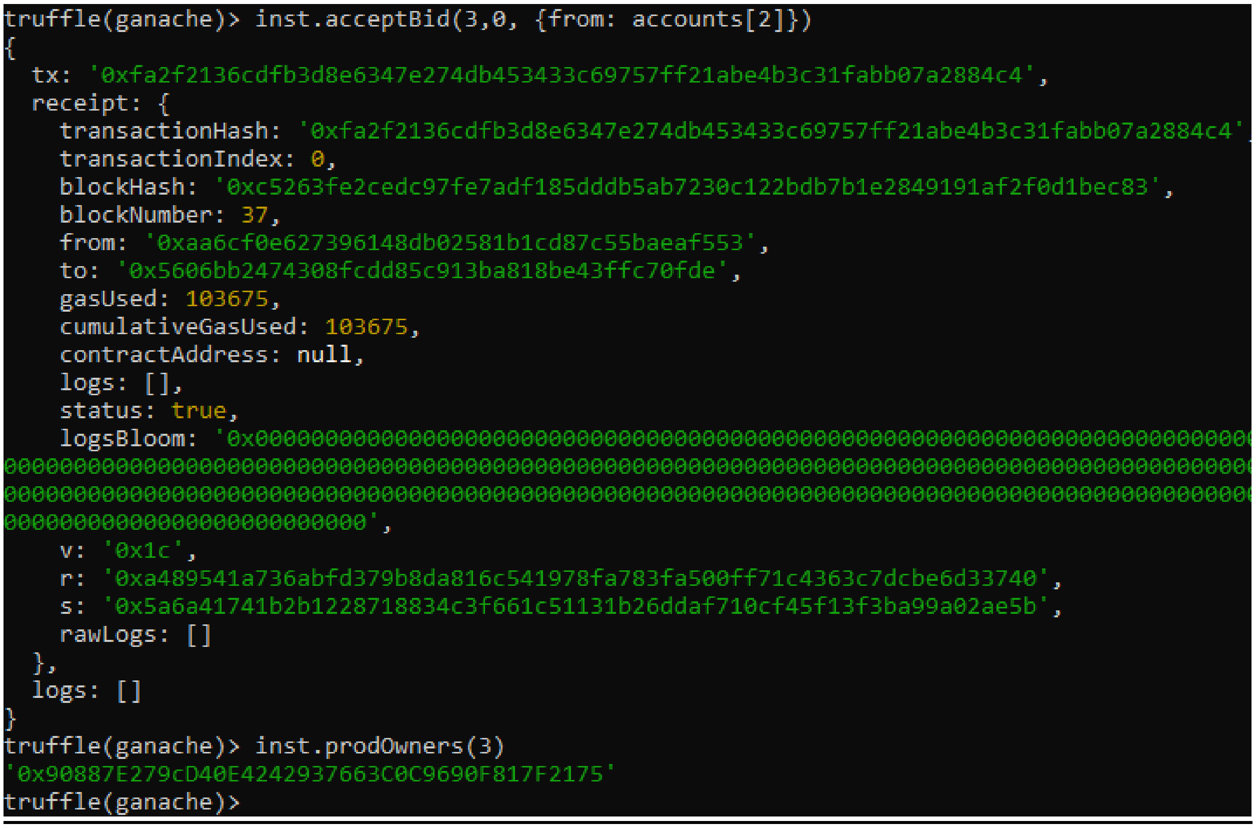
4. Accept bid (By product owner)

<u>Solidity Code:</u>



```
  function acceptBid(uint256 prodId,uint256 bidId)public {
     Product storage prod = userAllProds[msg.sender][prodId];
     address bidderAddr = prod.bids[bidId].bidUserAddress;
     prodOwners[prodId] = bidderAddr;
     userAllProds[bidderAddr][prodId] = prod;
     delete userAllProds[msg.sender][prodId];
 }
```

<u>Ganache:</u>



```
truffle(ganache)> inst.acceptBid(3,0, {from: accounts[2]})
{
  tx: '0xfa2f2136cdfb3d8e6347e274db453433c69757ff21abe4b3c31fabb07a2884c4',
  receipt: {
    transactionHash: '0xfa2f2136cdfb3d8e6347e274db453433c69757ff21abe4b3c31fabb07a2884c4',
    transactionIndex: 0,
    blockHash: '0xc5263fe2cedc97fe7adf185dddb5ab7230c122bdb7b1e2849191af2f0d1bec83',
    blockNumber: 37,
    from: '0xaa6cf0e627396148db02581b1cd87c55baeaf553',
    to: '0x5606bb2474308fcdd85c913ba818be43ffc70fde',
    gasUsed: 103675,
    cumulativeGasUsed: 103675,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000',
    v: '0x1c',
    r: '0xa489541a736abfd379b8da816c541978fa783fa500ff71c4363c7dcbe6d33740',
    s: '0x5a6a41741b2b1228718834c3f661c51131b26ddaf710cf45f13f3ba99a02ae5b',
    rawLogs: []
  },
  logs: []
}
truffle(ganache)> inst.prodOwners(3)
'0x90887E279cD40E4242937663C0C9690F817F2175'
truffle(ganache)>
```

In the last step account 2 has called acceptBid() function and accepted the bid of account 3 (who added the bid amount of 80) by entering his bid ID ‘0’.

So when prodOwners(3) function is called, we can see that owner of the product (id:3) has changed to account 3.