Online Auction System

PHASE - 4

TEAM MEMBERS:

First name: Venkat Saneet
Last name: Valluru
UB ID : 50353417
Email Id: vvalluru@buffalo.edu

First name: Teja Reddy
Last name: Alla
UB ID : 50409091
EmailId:tejaredd@buffalo.edu

ABSTRACT

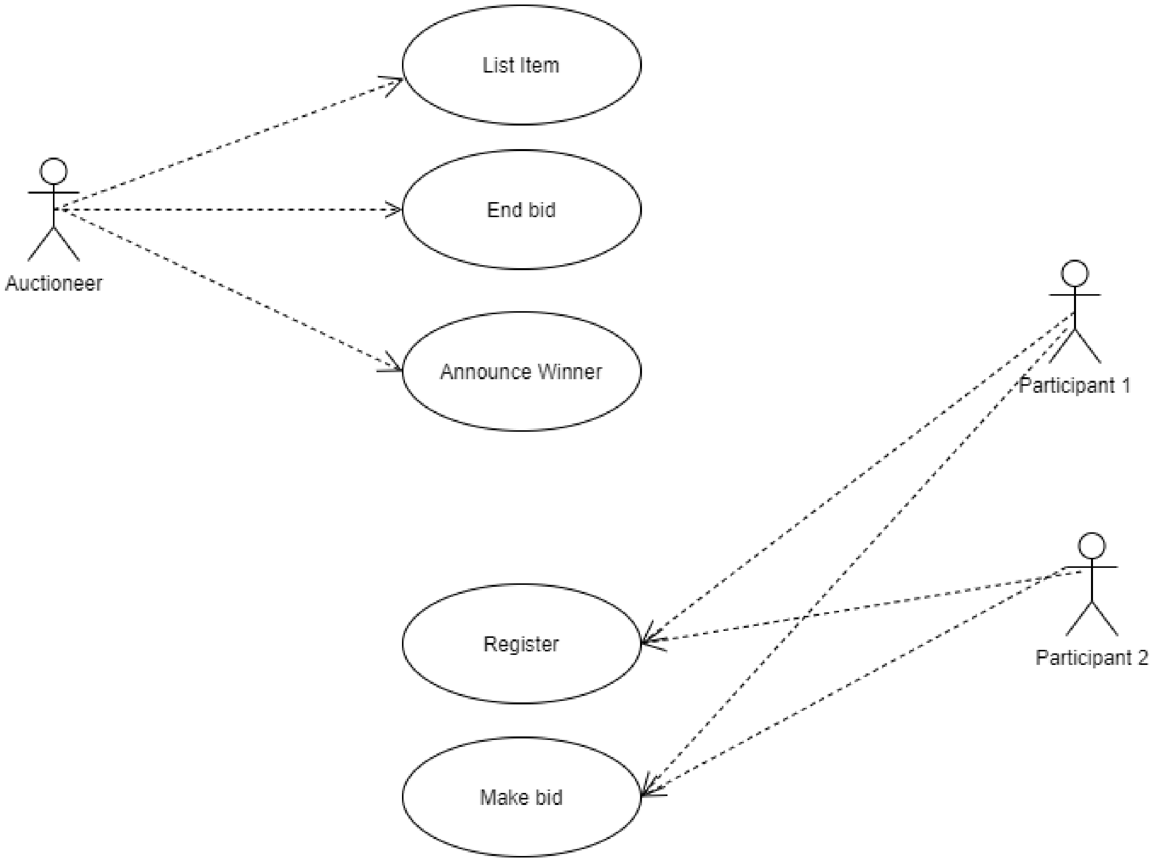
Online auction system is in principle same as the age old in-person auction system but implemented using the blockchain protocol. Unlike any other auction system, here the bids are placed using fungible tokens named TAI that are designed using the ERC20 standard and specially curated for smart online auction purposes. In smart online auction system, items are placed for auction by sellers, that can be bought by the buyer who places the highest bid. By designing a smart auction system following the Ethereum protocol we can eradicate the malpractices like bid-rigging, fake bidding that are rooted in traditional physical bidding processes. This smart auction system can be used by anyone who would like to auction off their assets like art, property, etc. As blockchain is a decentralized, distributed, digital ledger that is used to record transactions across many computers, the records cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network. This upholds the integrity of the auction and builds a sense of trust among the users.

PROCESS:

1. Each participant should register in order to participate in the bidding process.
2. Now, the Owner of the items will list the items for sale.
3. These listed items will be broadcasted to all the participants of the auction process.
4. Every one places their respective bids for the item they want to get.
5. After auction ends, the participant with the highest bid wins the round and gets the tokens in exchange for paying the owner.
6. All these transactions will occur using smart contracts and in the presence of a block chain ledger.

ARCHITECTURAL DIAGRAMS

1. USE CASE DIAGRAM



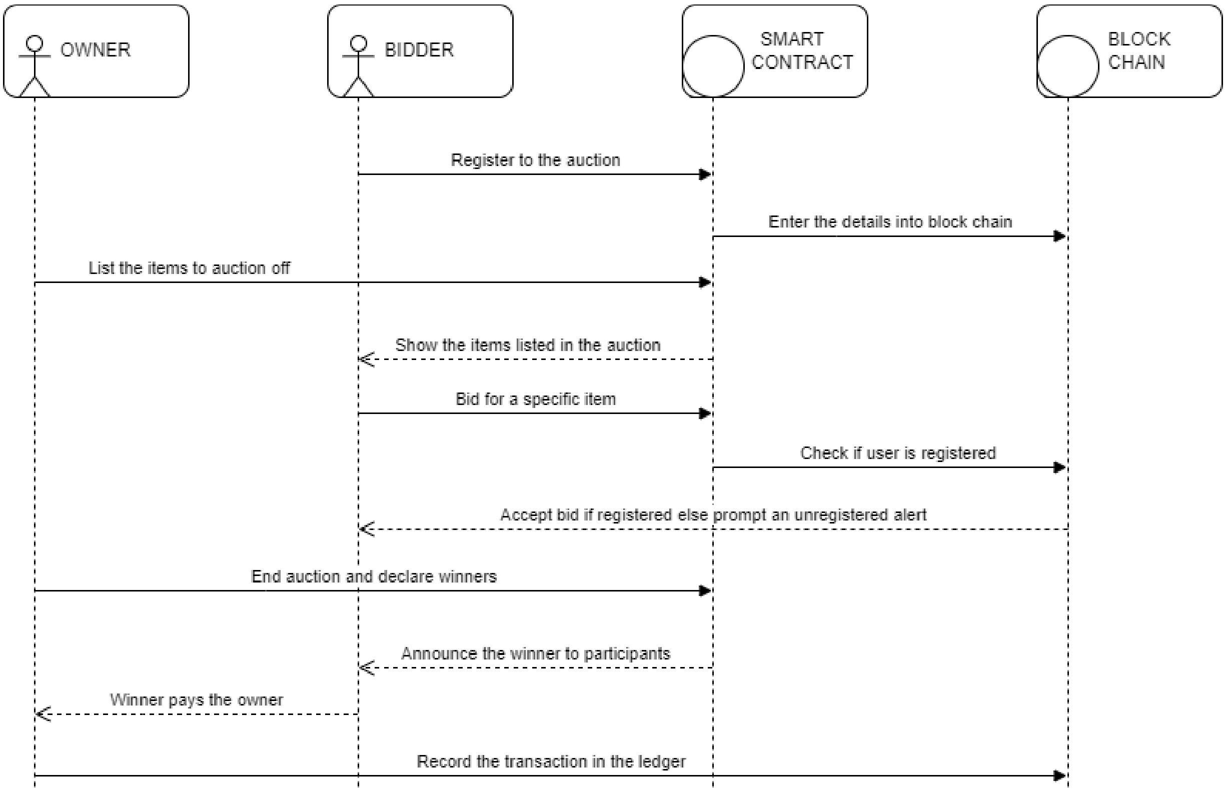
The Auctioneer can perform three tasks:

1. List Item
2. End Bid
3. Announce Winner

The Participant can perform the below actions:

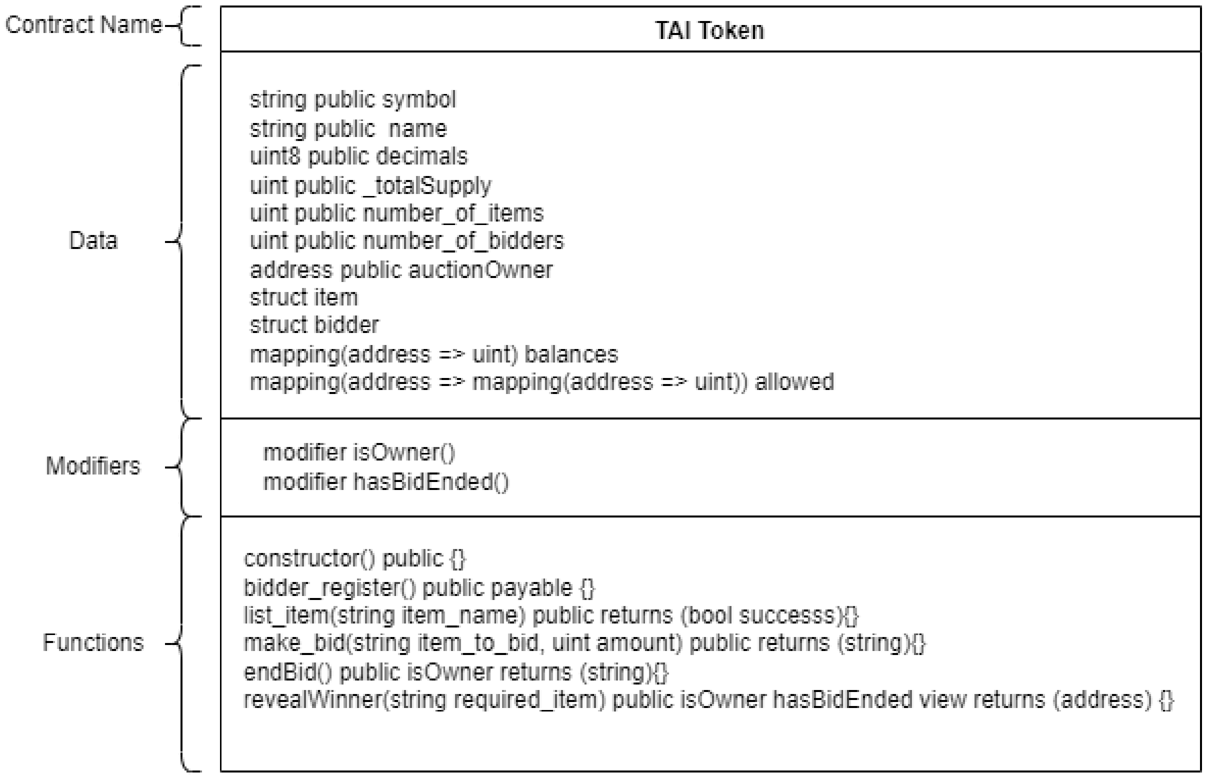
1. Register to the auction
2. Make a bid for the item

2. SEQUENCE DIAGRAM


OWNER
BIDDER
SMART CONTRACT
BLOCK CHAIN
Register to the auction
Enter the details into block chain
List the items to auction off
Show the items listed in the auction
Bid for a specific item
Check if user is registered
Accept bid if registered else prompt an unregistered alert
End auction and declare winners
Announce the winner to participants
Winner pays the owner
Record the transaction in the ledger

The sequence of the operations executed using the smart contract can be traced using a special diagram called sequence diagram. It helps us to find the flow of sequence of steps and the order in which each iteration takes place.

3. CONTRACT DIAGRAM:



TAI Token

```
string public symbol
string public  name
uint8 public decimals
uint public _totalSupply
uint public number_of_items
uint public number_of_bidders
address public auctionOwner
struct item
struct bidder
mapping(address => uint) balances
mapping(address => mapping(address => uint)) allowed
```

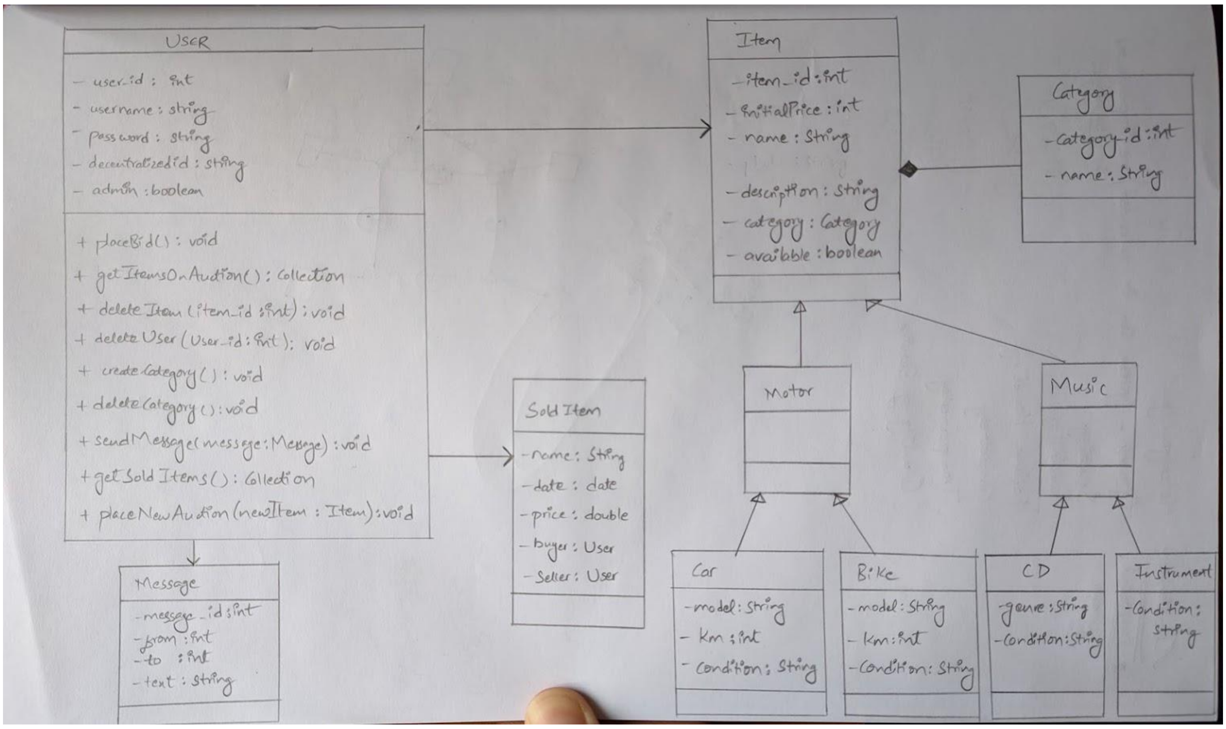
```
modifier isOwner()
modifier hasBidEnded()
```

```
constructor() public {}
bidder_register() public payable {}
list_item(string item_name) public returns (bool successs){}
make_bid(string item_to_bid, uint amount) public returns (string){}
endBid() public isOwner returns (string){}
revealWinner(string required_item) public isOwner hasBidEnded view returns (address) {}
```

Contract diagrams contains:

1. The name of the smart contract: TAI Token
2. The variables and their types, structures and mappings used in the smart contract.
3. Modifiers used in the smart contract.
4. Constructors and functions used in the contract.

4. UML DIAGRAM



This diagram helps to identify the relationship between each component and their dependencies.

ERC-20 PARAMETERS:

- **Name** - Trust And Integrity Token
- **Symbol** - TAI
- **Reasoning** - TAI stands for trust and Integrity. Since we want to design an online auction system that establishes trust and integrity, the symbol TAI will be the face of our system. It also helps in creating a positive intent among the users, if they are doubtful in participating in an unfamiliar online bidding environment.
- **Total Supply** – 100000.
- **Number of Decimal Places** - 2.

ISSUES ADDRESSED

- Breaking the age-old tradition of physical auction by implementing online auction.
- Secure and transparent auction system, where there is no place for foul play.
- Immutable bid history.
- Restricting fake bids (bids that the parties cannot afford).
- Tokenization of a real-life problem “Auction System”.

SPECIAL FUNCTIONS:

1. bidder_register():

Every bidder that plans to bid for any item should be registered before placing a bid, bidder_register function registers users so that the can place a bid.

- Number of bidders in the system is incremented by 1, whenever a user is registered.
- Bidder_id is assigned to every user which uniquely identifies them.
- This is a payable function, so the bidder is awarded with TAI tokens based on the ETH tokens which he used to send the message with.

2. list_item():

A token that is to be auctioned is listed using the list_item() function.

- It takes the token name as the input parameter.
- Every item listed in the auction has following attributes: name, owner, high_bid, high_bidder, high_bidder_id and bid count.
- When this function is invoked, the name of the listed item and the address of the person listing the item are stored.
- High_bid and bid_count parameters of the item are initialized with 0.

3. make_bid():

A bid is placed by the bidder, and the bid is only accepted when:

- The buyer has the amount specified in the bid.
- The bid amount is greater than minimum bid value.
- The present bid amount is greater than previous high bid.
- Whenever a successful bid is placed, the bid amount is temporarily transferred to the auction owner account.
- The previous high bid is transferred back to the previous high bidder from the auction owner account.

4. endBid():

After the completion of bidding process, the auction owner ends the bid of all the listed items by invoking this function, only the auction owner can invoke this function. After the completion of the auction the bid amount is transferred to the item owner from the auction owner account (Since the funds were temporarily transferred to auction owner whenever a higher bid is placed).

5. revealWinner():

The winner of the auction for that specific item is revealed by the owner of the auction using the reveal winner function where input the name of the item for which we need to reveal the winner. Thus, every other participant comes to know the details of the auction winner.

Assumptions:

1. All the participants know the process of bidding
2. Only unique items are auctioned and no fake listing.
3. The items won in the bidding process are sent to the winner in real time, hence the integrity of the person who lists the item matters.
4. Everything is recorded in sequence and updated in the immutable ledger.

Dependencies needed:

1. Ganache - Quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

2. Node.JS – to execute the JavaScript code

3. Truffle - A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM)

4. Metamask - MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications.

5. ExpressJS – It is designed for building web applications and APIs.

6. Nodemon - nodemon is a tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected.

7. Web3 - web3.js is a collection of libraries that allow you to interact with a local or remote ethereum node using HTTP, IPC or WebSocket.

Steps to executes:

- Initialize a new truffle project to develop our project using **truffle init.**

- We install the dependencies using **npm install**.

- Then, navigate to the path of the smart contract folder.

- **Truffle compile** to compile the smart contract.

- **Truggle migrate** to deploy the smart contract.

- Now the smart contract is successfully compiled and deployed on to the block chain (Ganache)

- Every time we deploy the smart contract, we need to change the contract address in the base.js code to ensure that we are interacting with the correct address.

- Now the web interface can be accessed through localhost:3000

- First, anyone who wants to take part in the auction needs to register with some initial ether.

- Then the participant can list an item in the auction

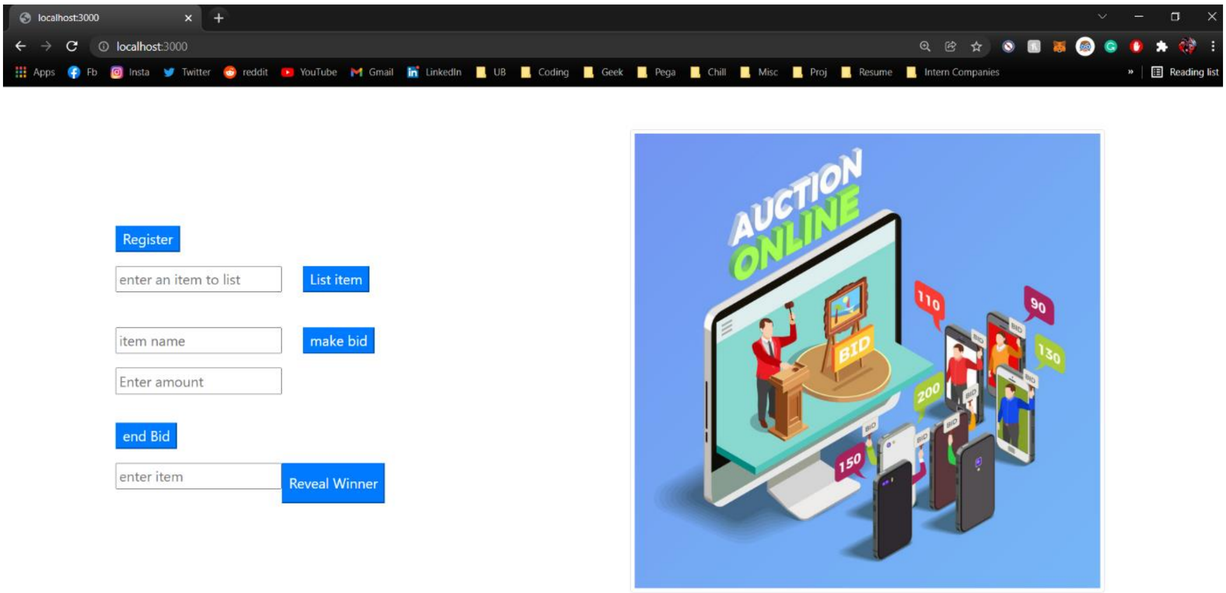
- This way many bidders register and can list their own items

- Then whoever wants to bid needs to specify the item name and the amount to bid. If it is a valid bid, it will be accepted.

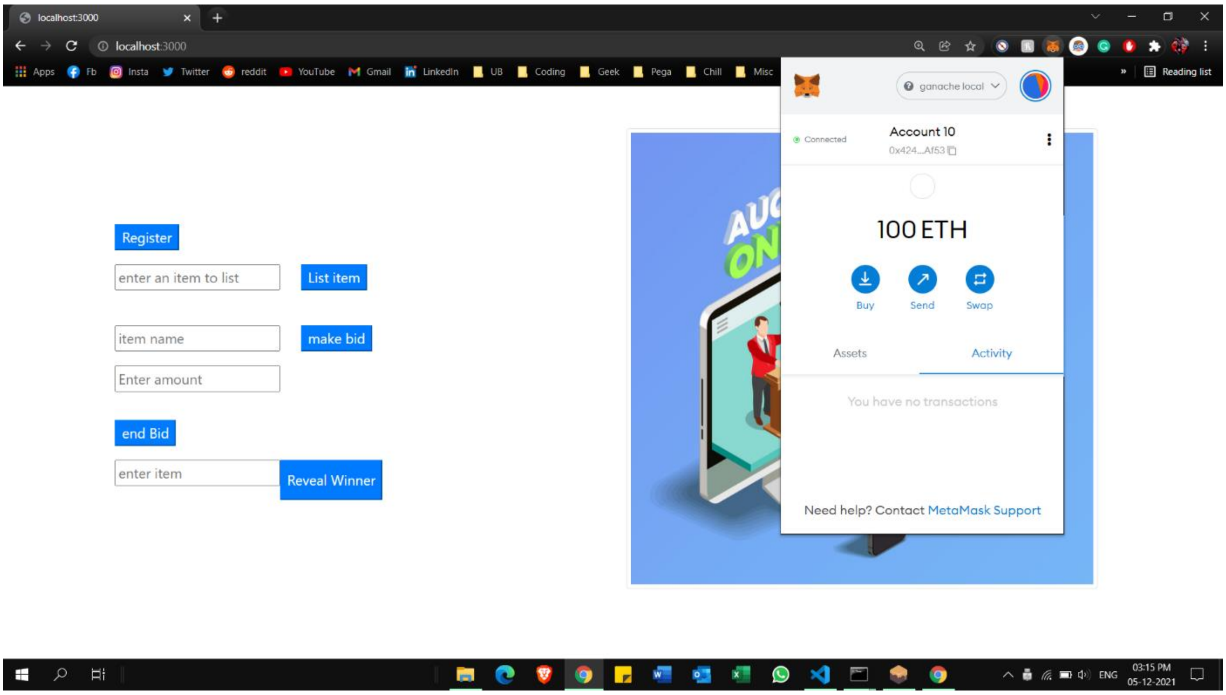
- After all the bids are placed and the auction is ended, the owner will reveal the winner of each item in the auction.

Screenshots:

1. Initial web interface

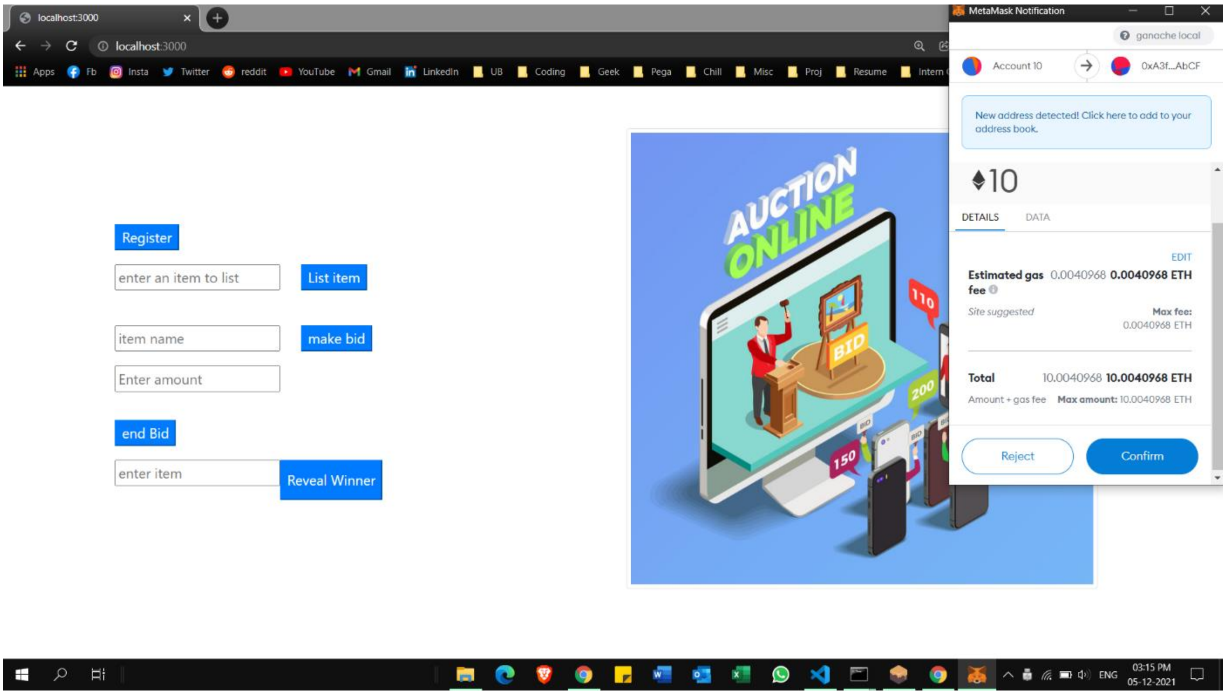


2. Now Account 10 will register to participate in the bid using 10 ether.



You have no transactions

Need help? Contact MetaMask Support



New address detected! Click here to add to your address book.

10

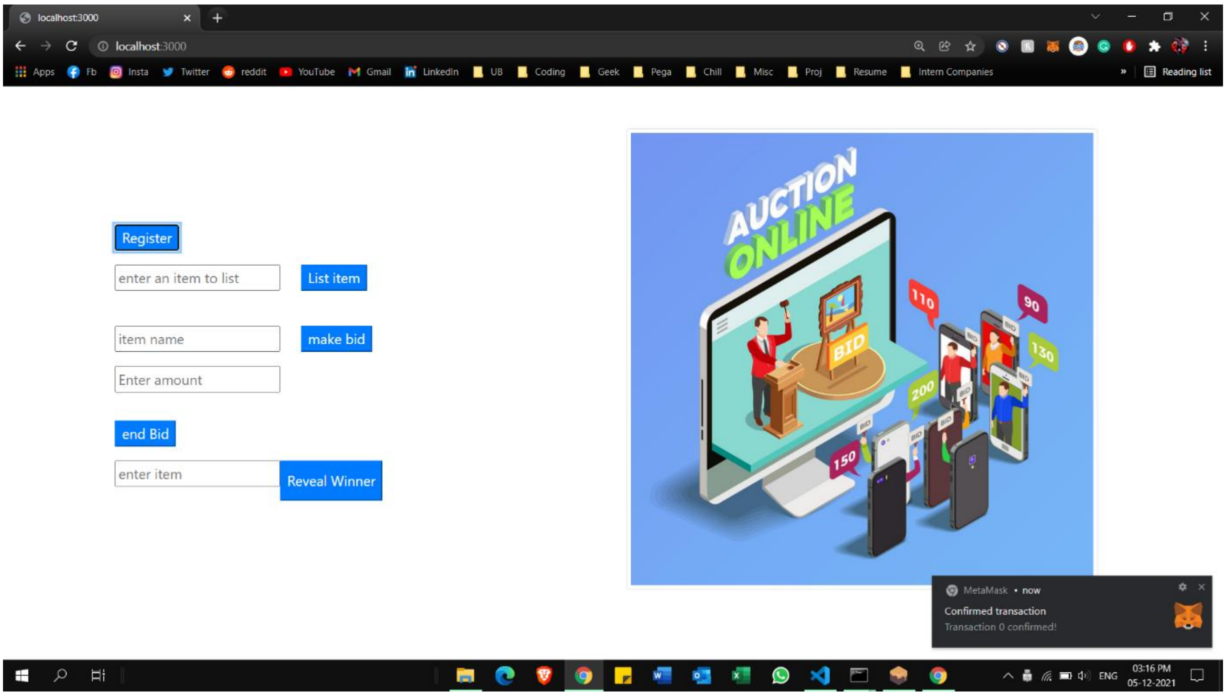
Estimated gas fee 0.0040968 **0.0040968 ETH**

Site suggested **Max fee:** 0.0040968 ETH

Total 10.0040968 **10.0040968 ETH**

Amount + gas fee **Max amount:** 10.0040968 ETH

03:15 PM 05-12-2021



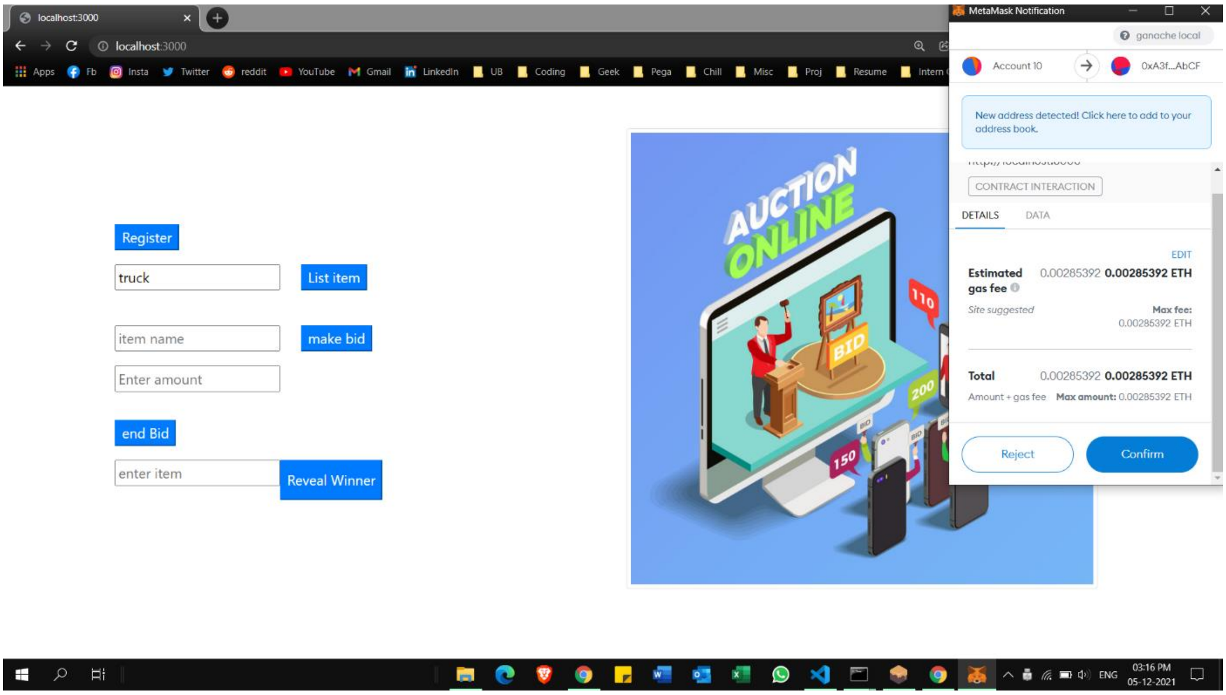
MetaMask • now

Confirmed transaction

Transaction 0 confirmed!

03:16 PM 05-12-2021

3.Account 10 will list an item “truck” to auction off



truck

item name

Enter amount

enter item

ganache local

Account 10 → 0xA3f...AbCF

New address detected! Click here to add to your address book.

CONTRACT INTERACTION

Estimated gas fee 0.00285392 **0.00285392 ETH**

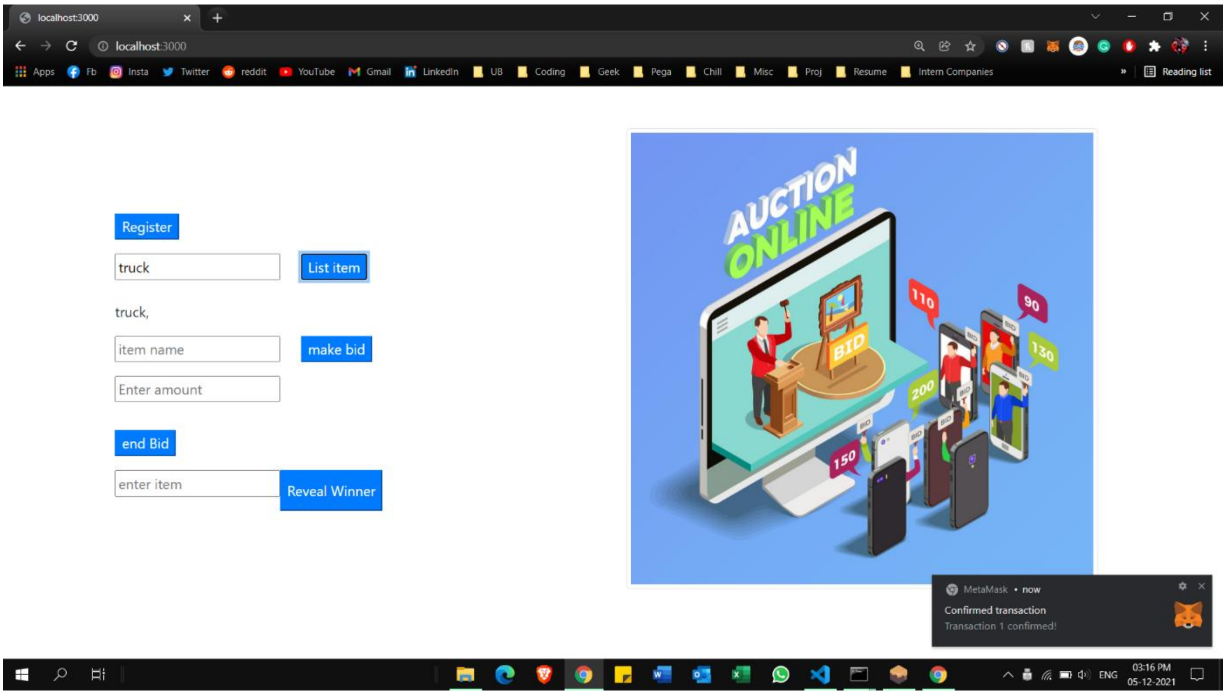
Site suggested Max fee: 0.00285392 ETH

Total 0.00285392 **0.00285392 ETH**

Amount + gas fee **Max amount:** 0.00285392 ETH

03:16 PM 05-12-2021

4. As soon as an item is listed for auction, all the available items to bid are listed as seen in the below screenshot



truck

truck,

item name

Enter amount

enter item

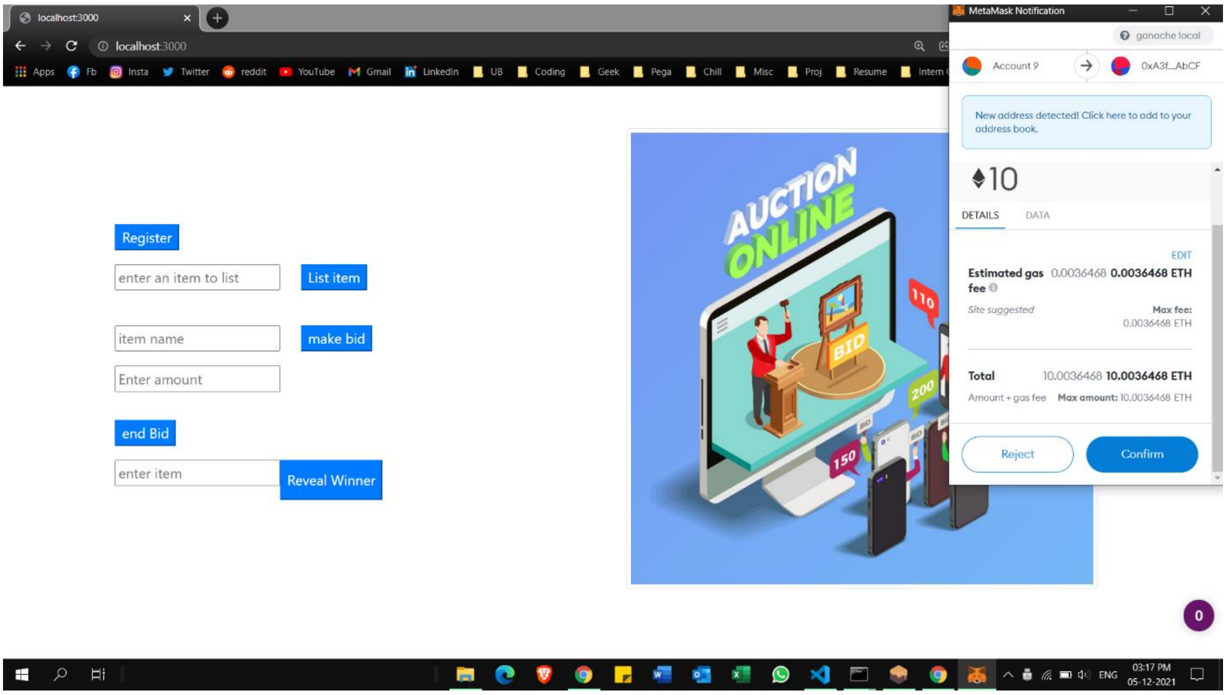
MetaMask • now

Confirmed transaction

Transaction 1 confirmed!

03:16 PM 05-12-2021

5. Now Account 9, Account 8 and Account 7 are trying to register to participate in the auction



MetaMask Notification

ganache local

Account 9 → 0xA3f...AbCF

New address detected! Click here to add to your address book.

10

DETAILS DATA

EDIT

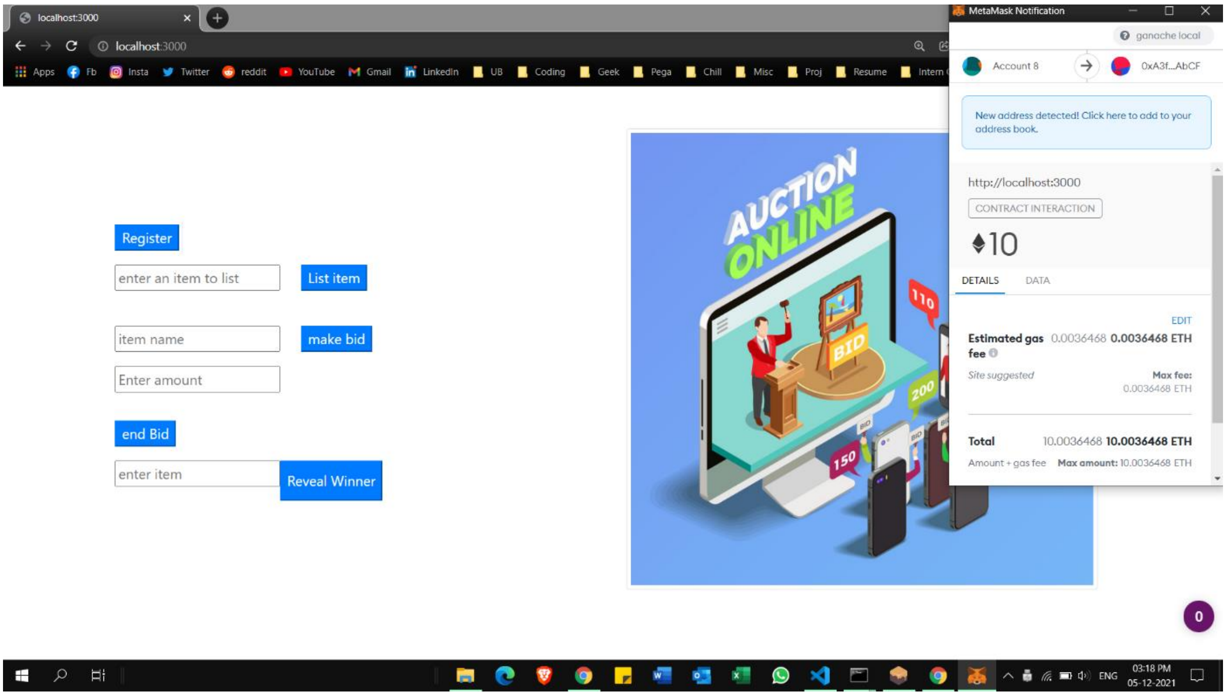
Estimated gas fee 0.0036468 **0.0036468 ETH**

Site suggested **Max fee:** 0.0036468 ETH

Total 10.0036468 **10.0036468 ETH**

Amount + gas fee **Max amount:** 10.0036468 ETH

Reject Confirm



MetaMask Notification

ganache local

Account 8 → 0xA3f...AbCF

New address detected! Click here to add to your address book.

http://localhost:3000

CONTRACT INTERACTION

10

DETAILS DATA

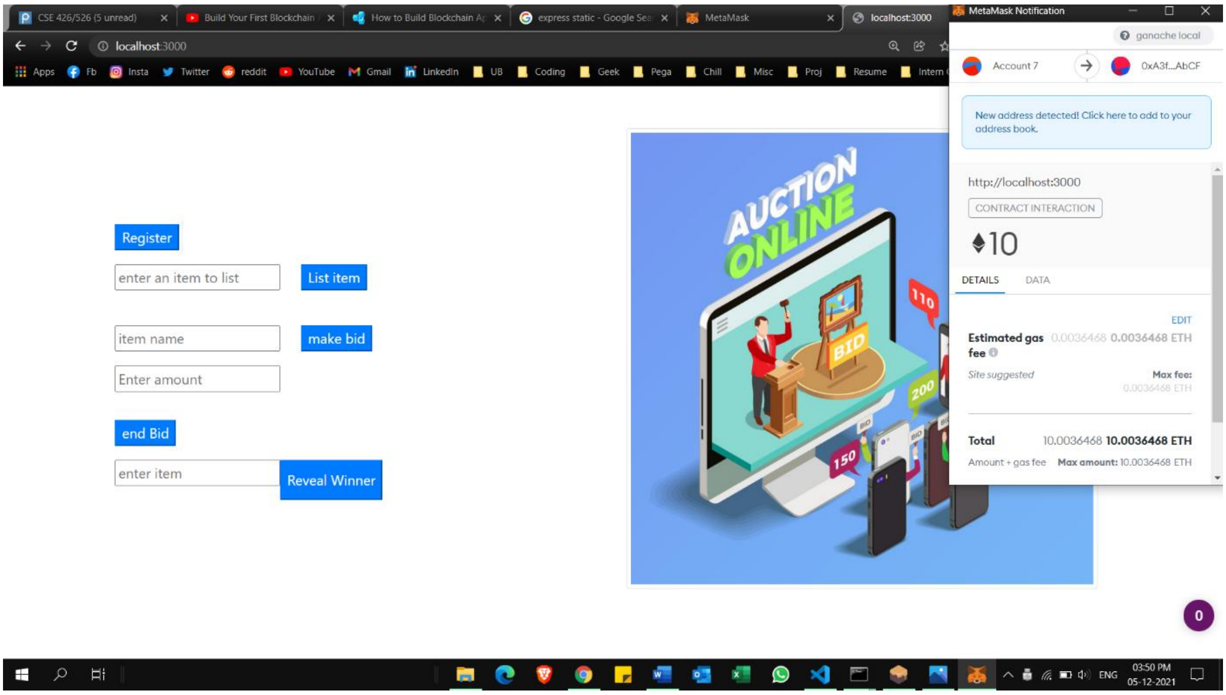
EDIT

Estimated gas fee 0.0036468 **0.0036468 ETH**

Site suggested **Max fee:** 0.0036468 ETH

Total 10.0036468 **10.0036468 ETH**

Amount + gas fee **Max amount:** 10.0036468 ETH



New address detected! Click here to add to your address book.

http://localhost:3000

CONTRACT INTERACTION

10

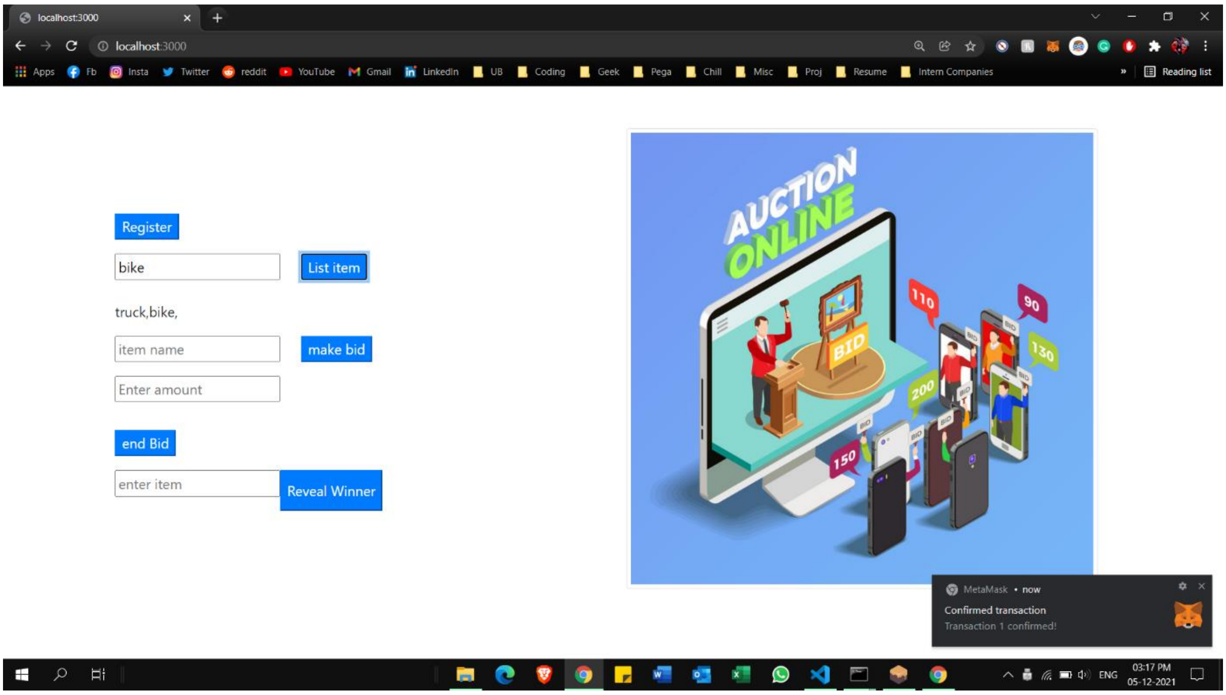
Estimated gas fee 0.0036468 0.0036468 ETH

Site suggested Max fee: 0.0036468 ETH

Total 10.0036468 10.0036468 ETH

Amount + gas fee Max amount: 10.0036468 ETH

6. Account 7 lists an item bike to auction off



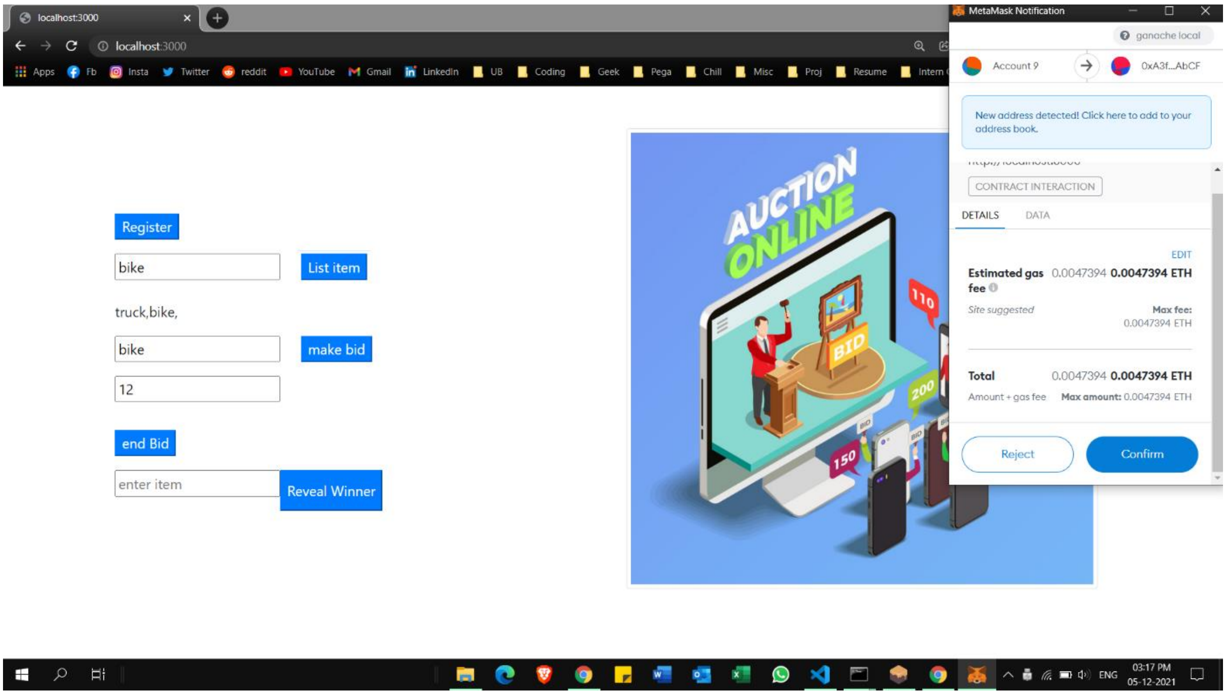
bike

truck,bike,

Confirmed transaction

Transaction 1 confirmed!

7. Now account 9 bids for the item bike with 12 tokens



bike

truck,bike,

bike

12

enter item

Account 9 → 0xA3f...AbCF

New address detected! Click here to add to your address book.

Estimated gas fee 0.0047394 0.0047394 ETH

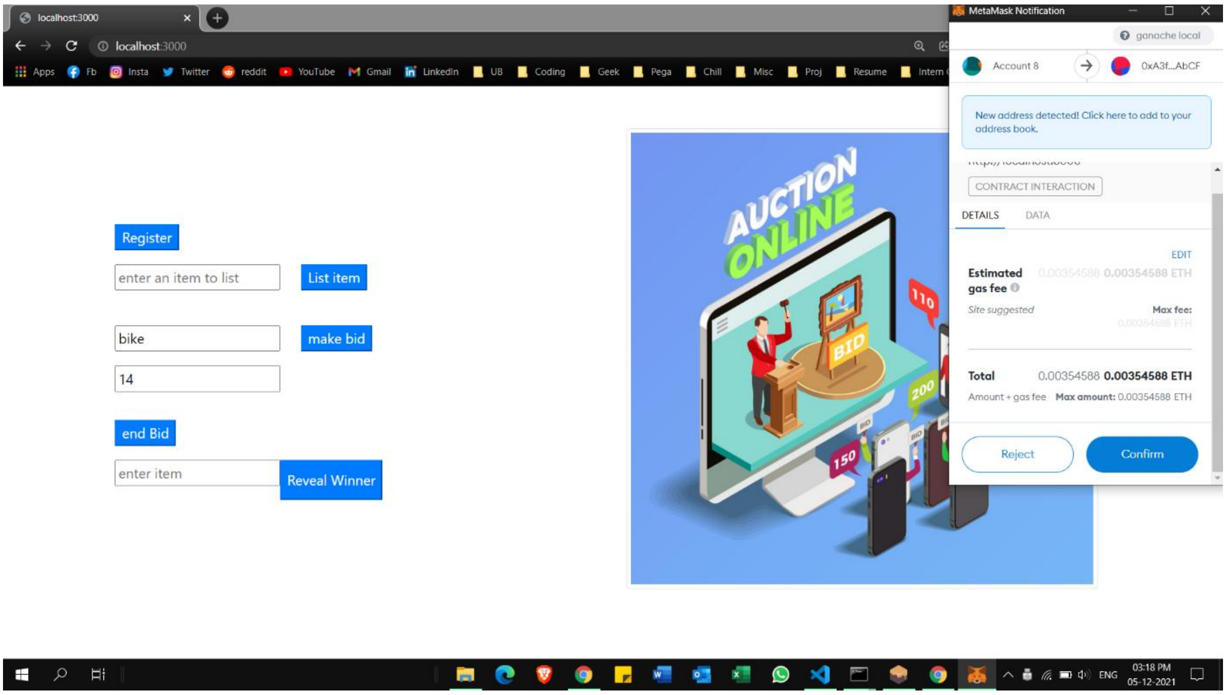
Site suggested Max fee: 0.0047394 ETH

Total 0.0047394 0.0047394 ETH

Amount + gas fee Max amount: 0.0047394 ETH

03:17 PM 05-12-2021

8. Now account 8 bids for the item bike with 14 tokens



enter an item to list

bike

14

enter item

Account 8 → 0xA3f...AbCF

New address detected! Click here to add to your address book.

Estimated gas fee 0.00354588 0.00354588 ETH

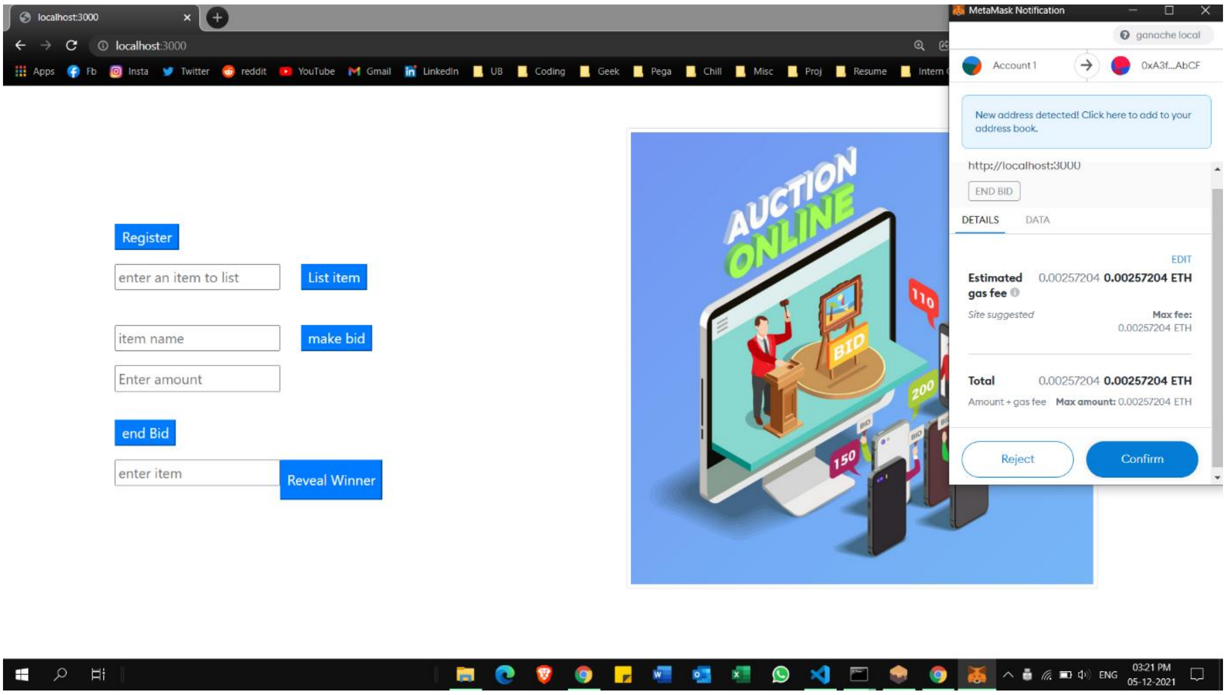
Site suggested Max fee: 0.00354588 ETH

Total 0.00354588 0.00354588 ETH

Amount + gas fee Max amount: 0.00354588 ETH

03:18 PM 05-12-2021

9. Account 1, who is the auction owner will end the bid and reveal winners for the item bike.



New address detected! Click here to add to your address book.

http://localhost:3000

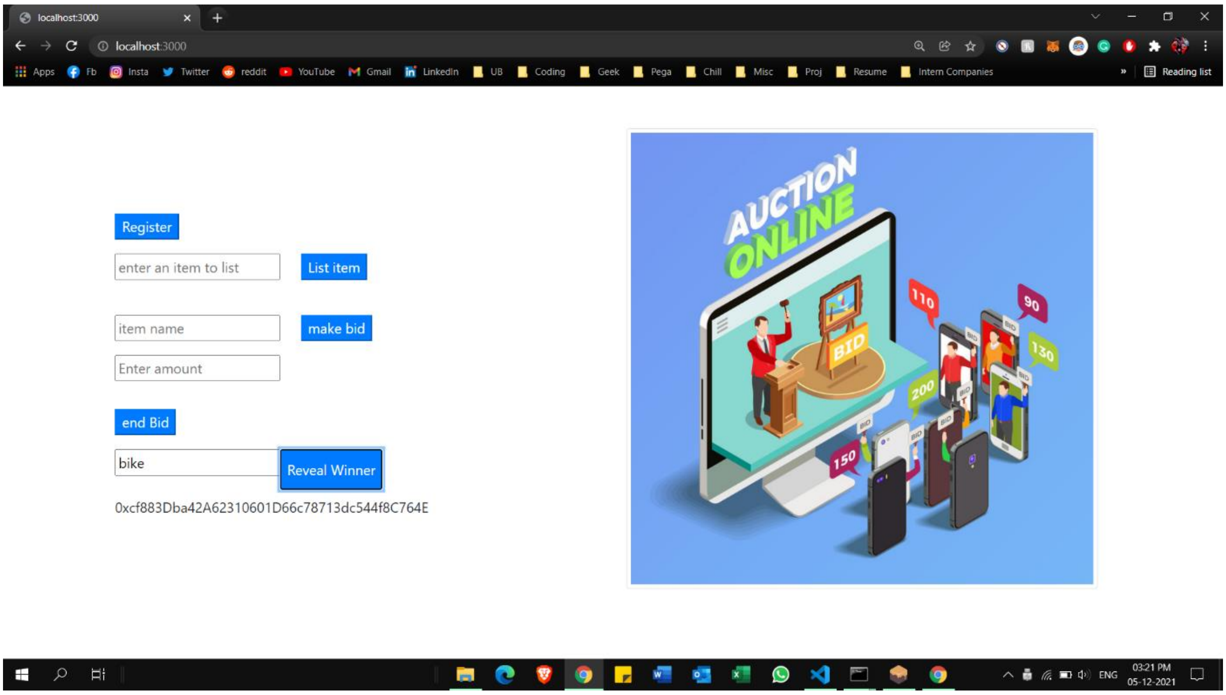
END BID

Estimated gas fee 0.00257204 **0.00257204 ETH**

Site suggested — Max fee: 0.00257204 ETH

Total 0.00257204 **0.00257204 ETH**

Amount + gas fee — Max amount: 0.00257204 ETH



bike

0xcf883Dba42A62310601D66c78713dc544f8C764E

10. “**0xcf883Dba42A62310601D66c78713dc544f8C764E**” which is the address of **Account 8** is the winner of the item bike.

And as Account bid for bike with 14 tokens which is greater that Account 9’s bid of 12 tokens for bike.

Thus Account 8 has won the bidding for the item bike.