Building a Privacy-Preserving Blockchain-Based Bidding System: A Crypto Approach

Blockchain has inbuilt support for data unchangeable and encryption which is making Blockchain more secure and this advantages forcing application developers to migrate towards Blockchain. Online Bidding is one of the application where bidders will bid for particular tenders and some malicious internal employees will change bidding or tender details to make their favourable person to win bidding and this manipulation will cost huge loss to the bidding system.

In propose paper author is employing Blockchain technology on bidding system to avoid such manipulation and to allow full privacy for both seller and bidder Deniable Encryption technique is applying which allow only those seller and bidder to decrypt data which got encrypted using same seller and bidder. So Deniable Encryption on top of Blockchain will provide full privacy to seller and bidder data.

To verify biding data author is using Chameleon hash code which generate unique code on bidding data and if data not changed then same Chameleon Hash Code will be generated and data verification will be successful and if data changed then different hash code will be generated which indicate data is altered. So by applying Chameleon we can verify bidding data.

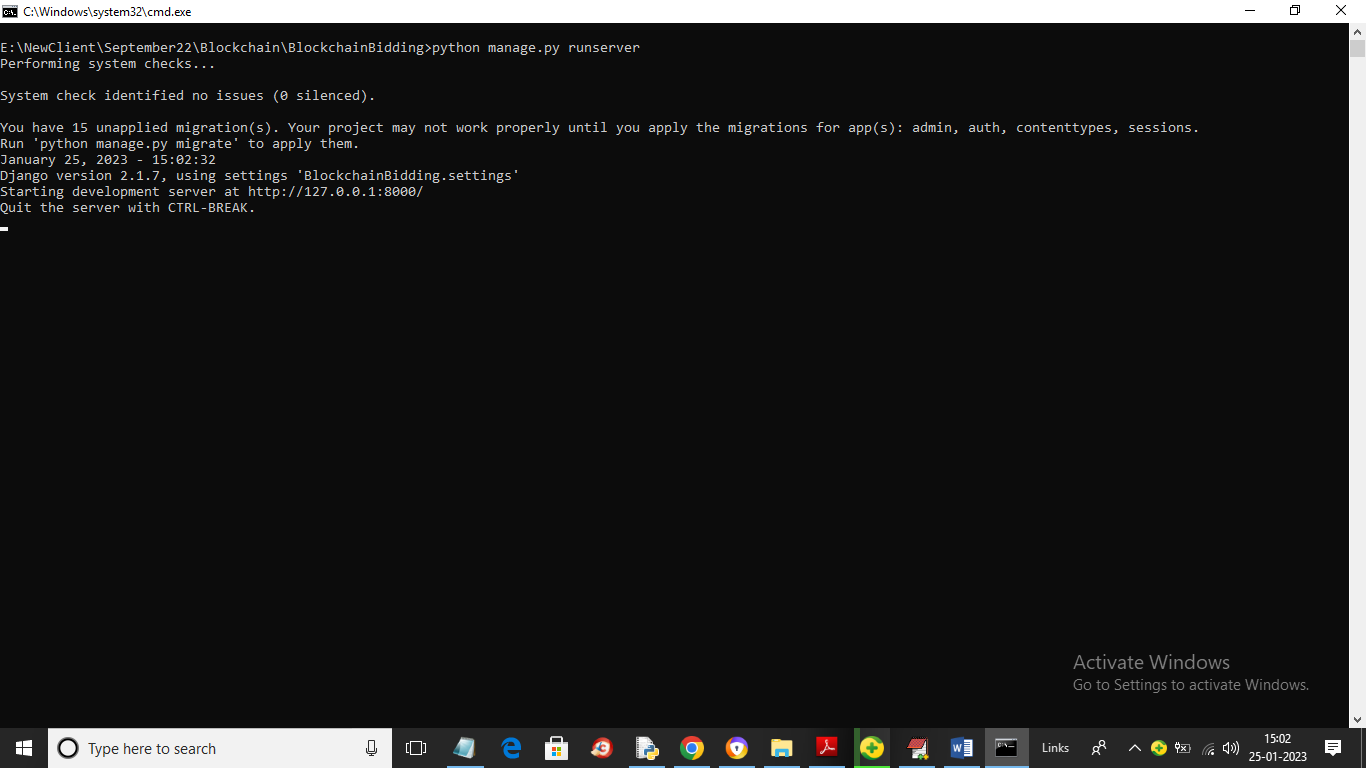
Blockchain-based bidding systems are becoming increasingly popular nowadays. Due to the properties of Blockchain, bidding records are unchangeable. With existing encryption techniques, these bidding records can only be shared by the bidder and the seller. Although this scenario sounds secure, it does not consider a coercion case. A powerful coercer may force the bidding system to open the records stored on the Blockchain, and the system loses privacy. To solve this problem, in this paper, in propose paper a new encryption scheme called deniable matchmaking encryption (DME). This new encryption scheme provides deniability not only for the message, but also for the identities. We use the chameleon hash function to make fake message and fake identities indistinguishable from the real message and the real identities. Therefore, the bidding system can use fake information to answer the coercer, and user privacy is kept by the Blockchain-based bidding system.

Extension Concept: in propose paper author is allowing only seller and bidder to view winning notifications and other bidders will not know about winning bidder so as extension we are modifying DME algorithm to allow decryption and view by all the other bidders who bid on same tender about bidding winner. So winner and other bidders can decrypt and view all the bidding details and winner details

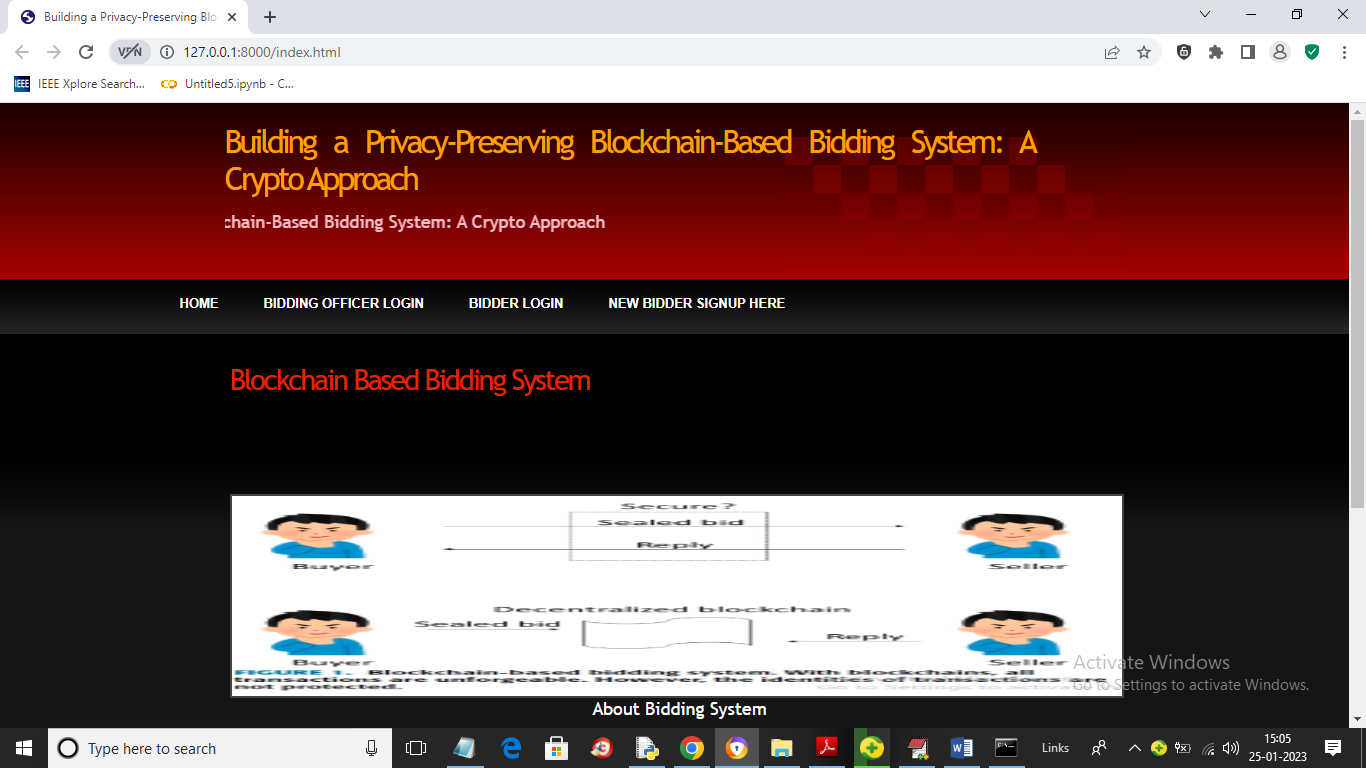
To implement this project we have designed following modules

1. Bidding Officer Login: bidding officer will login to application using username and password as ‘admin’ and ‘admin’ and then create tender with initial bidding amount. This user will evaluate bidding to select bidder with LOWEST amount and then will run ‘Winner Selection’ module to select winner.
2. Bidder Login: Bidder can signup with the application and then login and then view all tenders and bidding and can bid for desired tender. Tender status can be view by clicking on ‘View Status’ link which will display all tender, bidding and winner details.

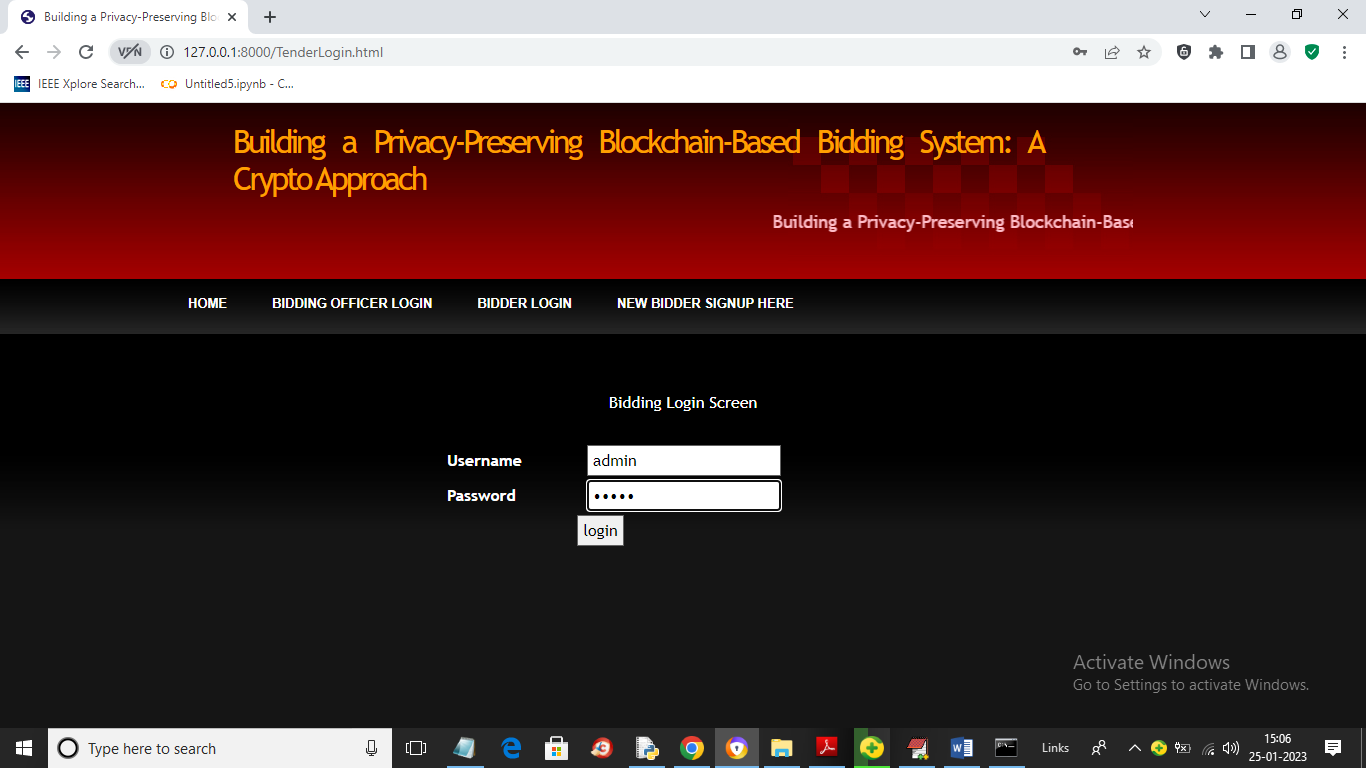
To run project double click on ‘run.bat’ file to start python server and get below screen



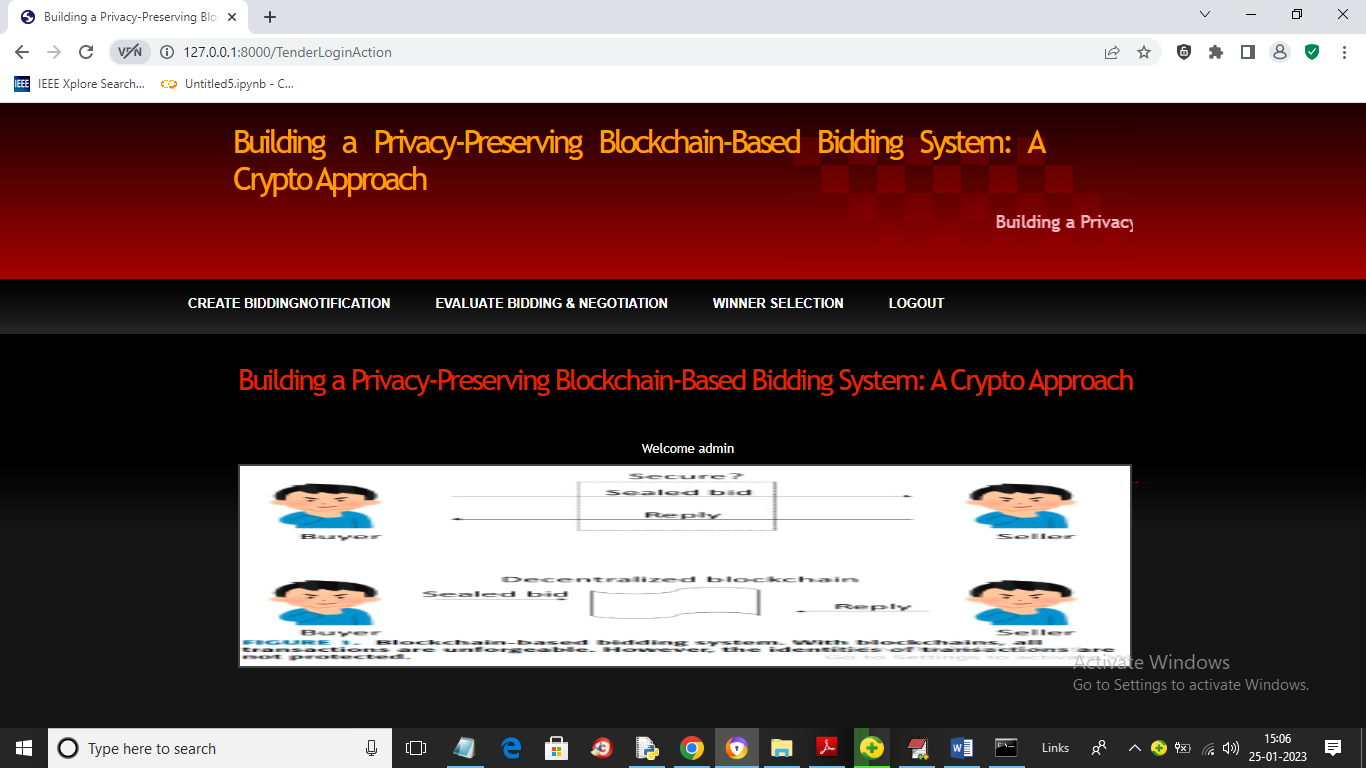
In above screen python web server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below page



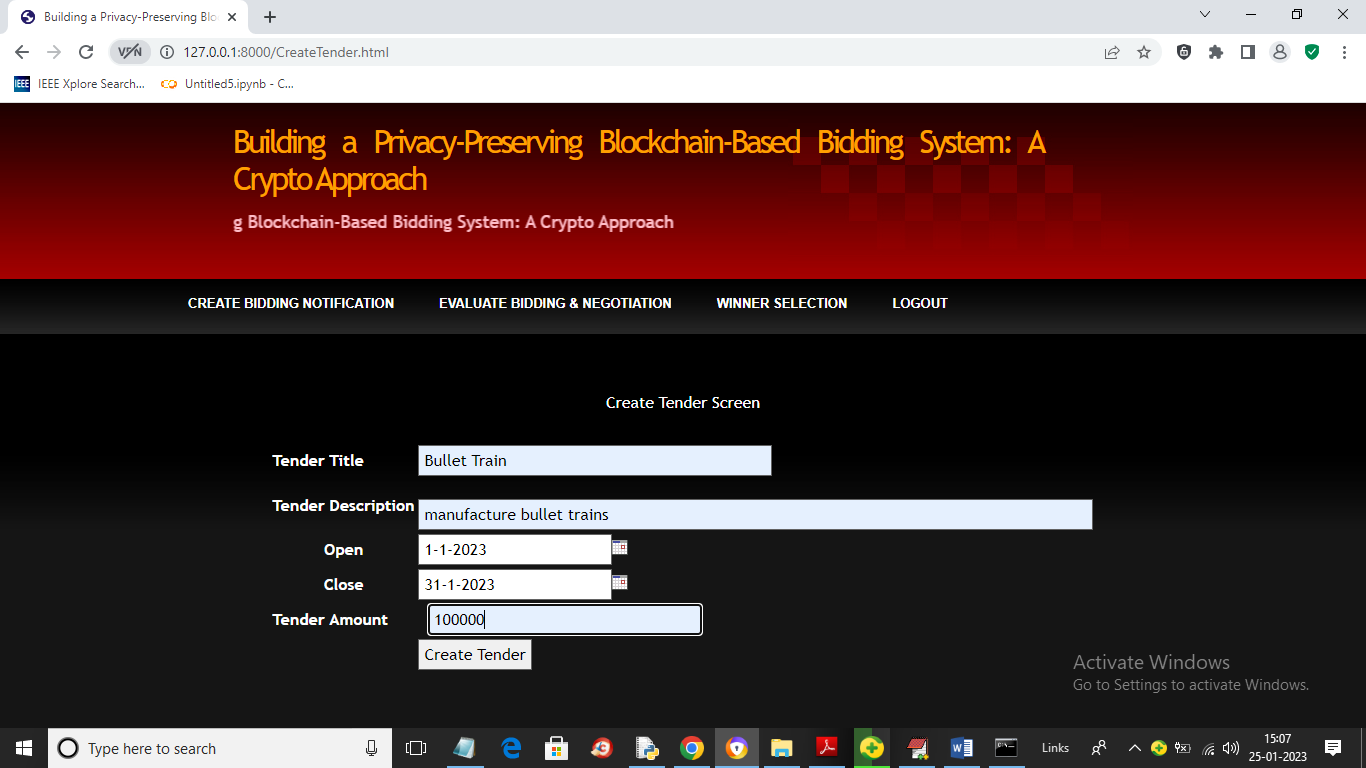
In above screen click on ‘Bidding Officer Login’ link to get below page



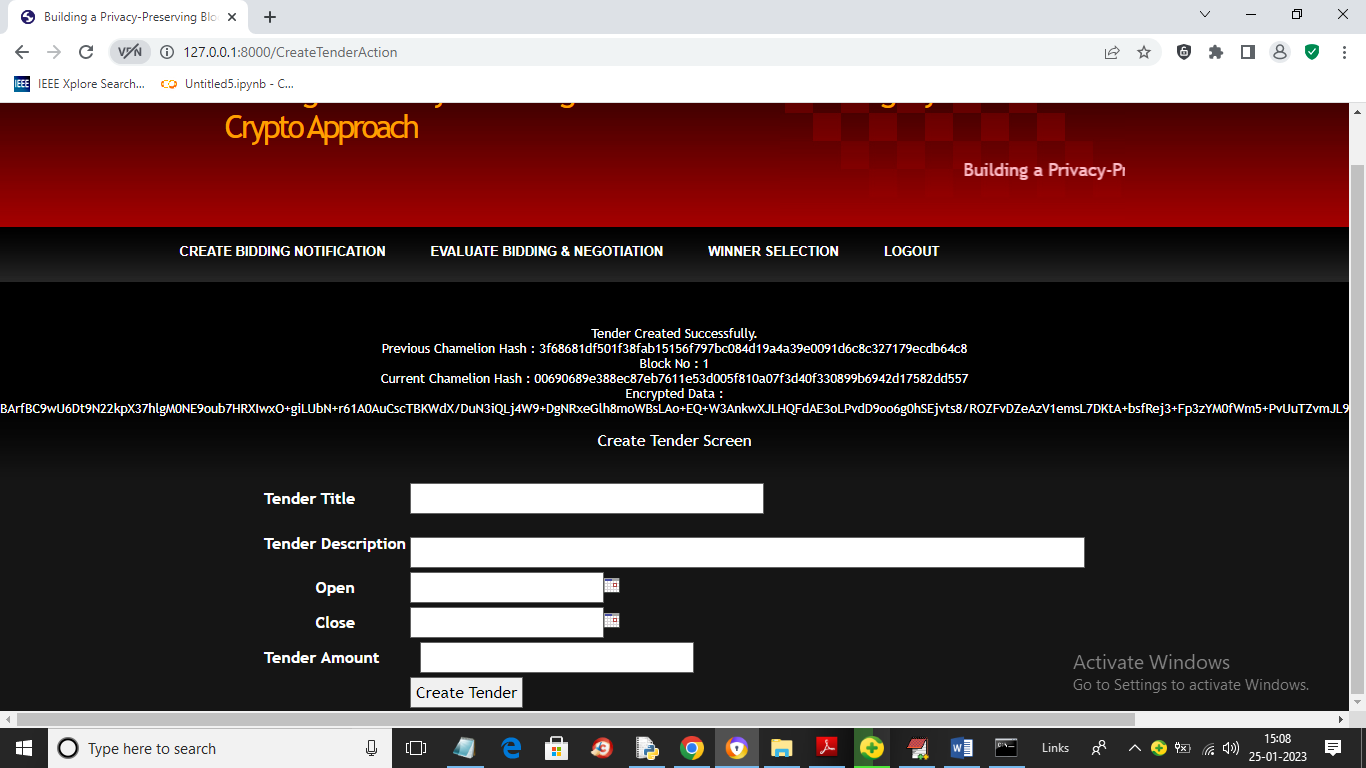
In above screen bidding officer is login and after login will get below page



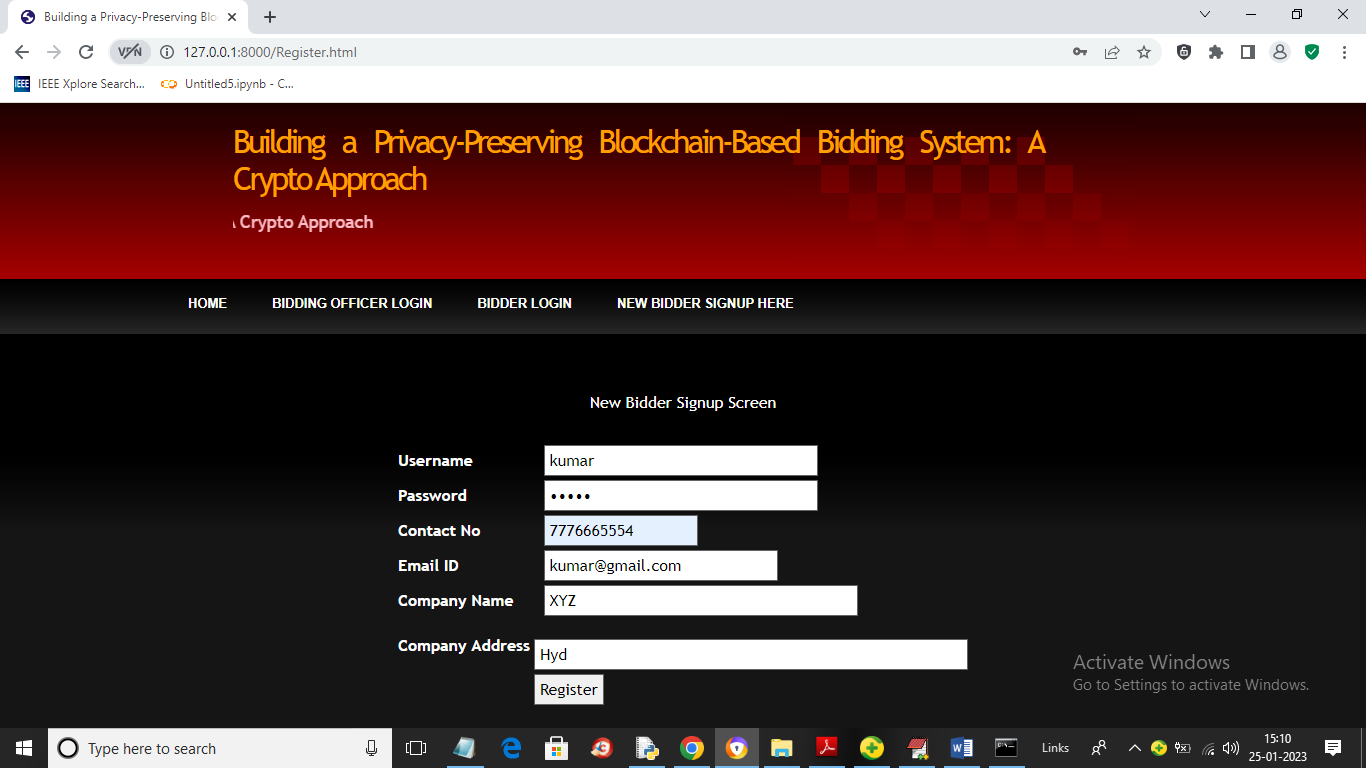
In above screen bidding officer can click on ‘Create Bidding Notification’ link to create tender like below screen



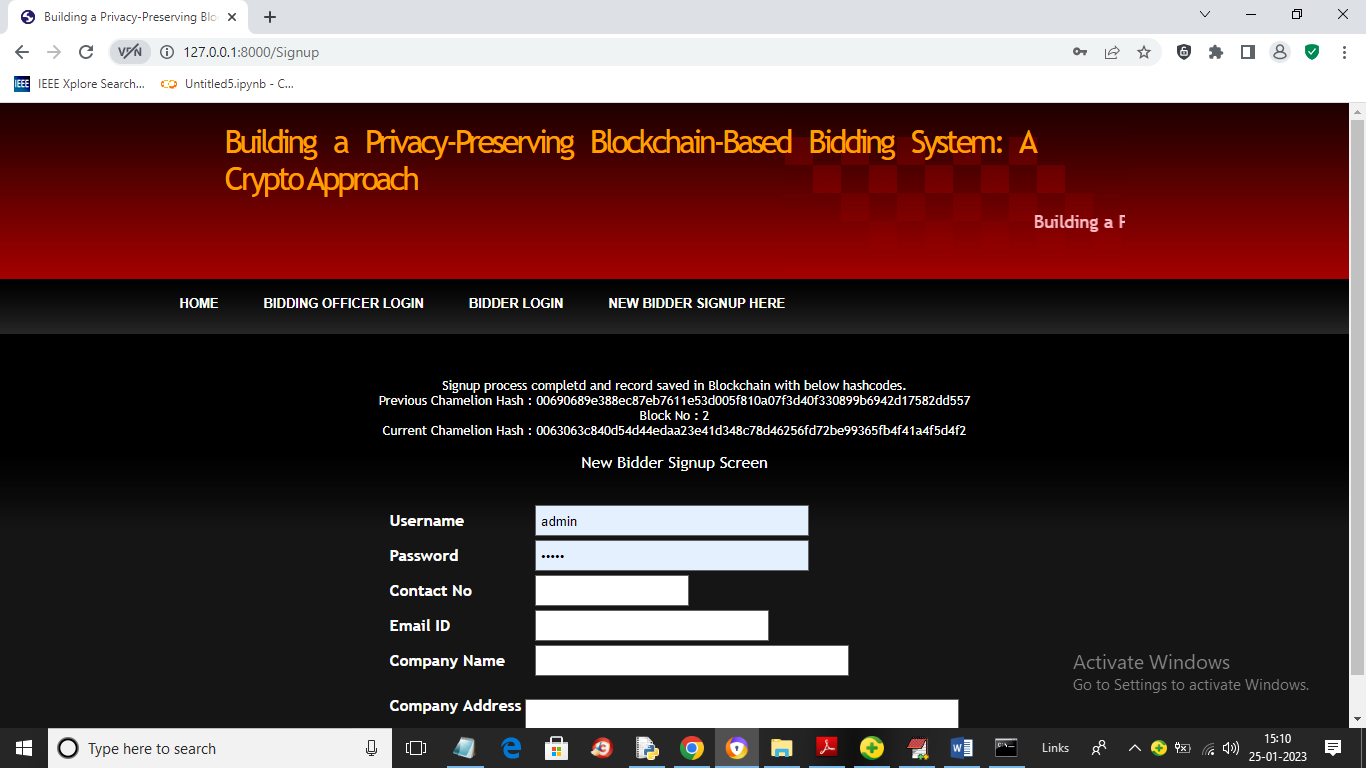
In above screen we are creating one tender with start and end date with initial tender amount and now press button to save data in Blockchain and will get below output



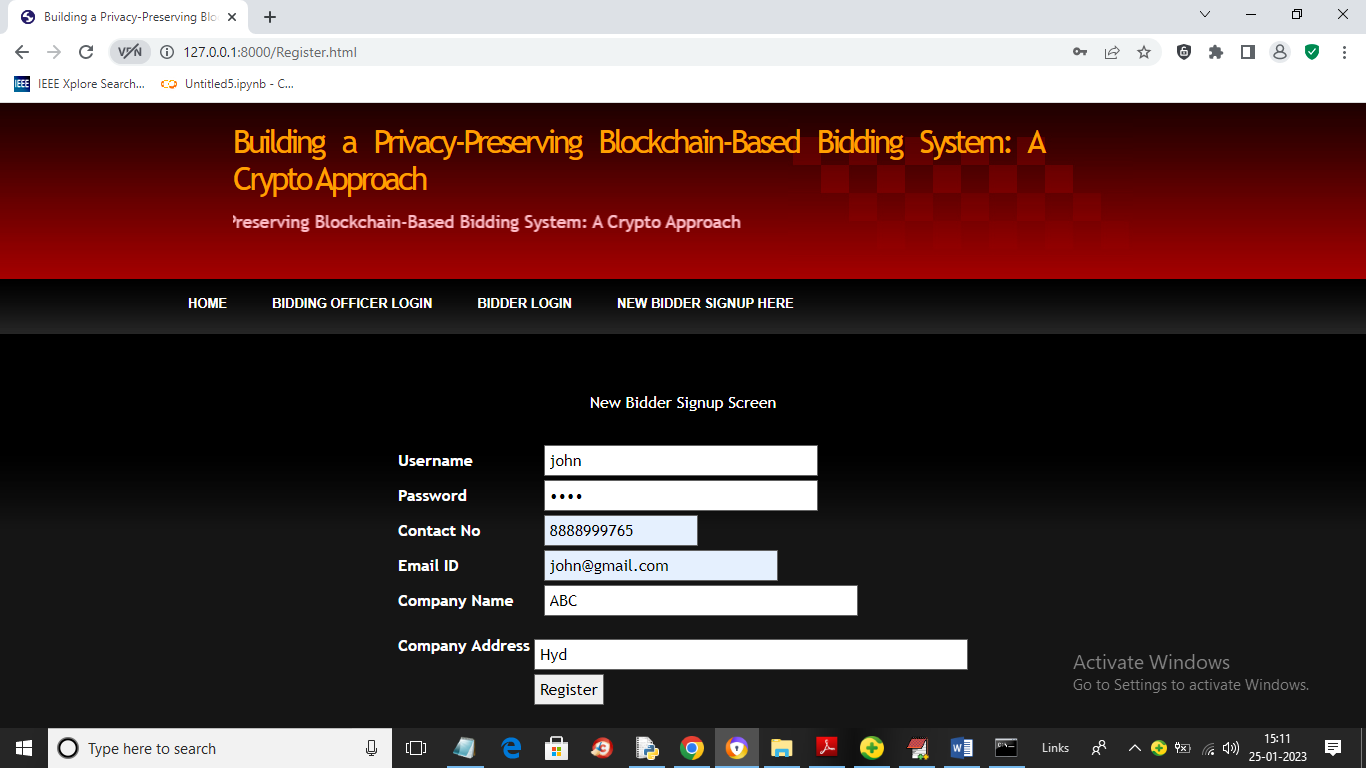
In above screen in first line we can see ‘Tender Created’ and then we can see generated previous and current block Chameleon hash code and then displaying encrypted data and now logout and signup and login bidders to bid for above tender



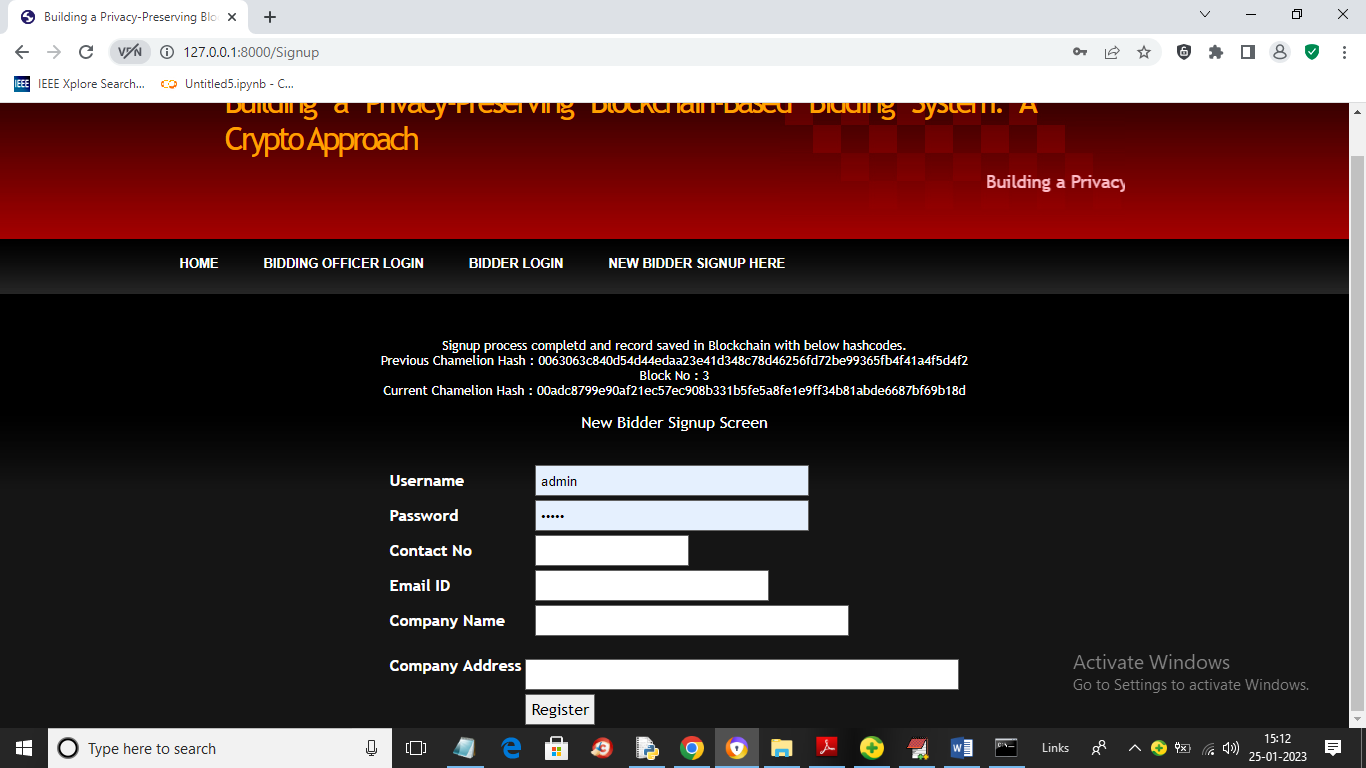
In above screen one user is signing up and then press button to get below page



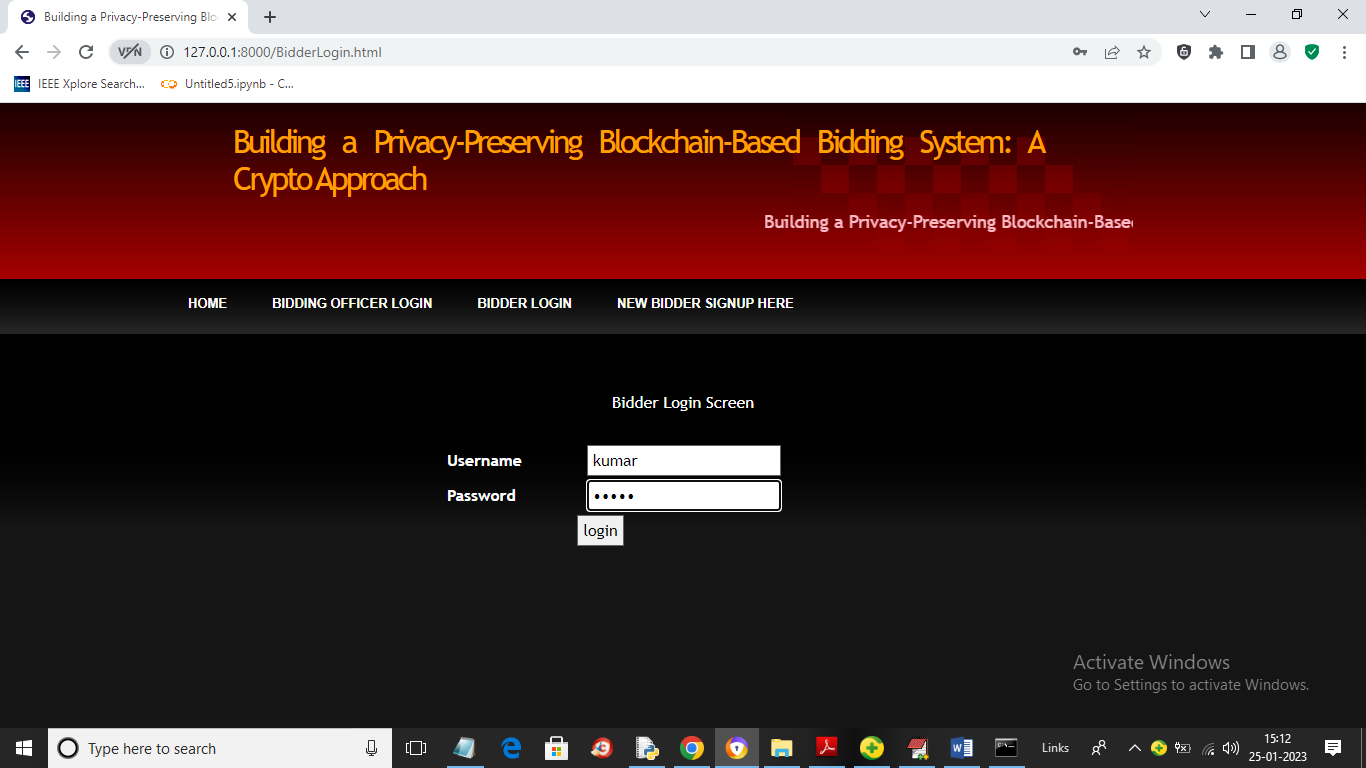
In above screen we can see signup task completed and we can see hash code returned by Blockchain and similarly design any number of bidders



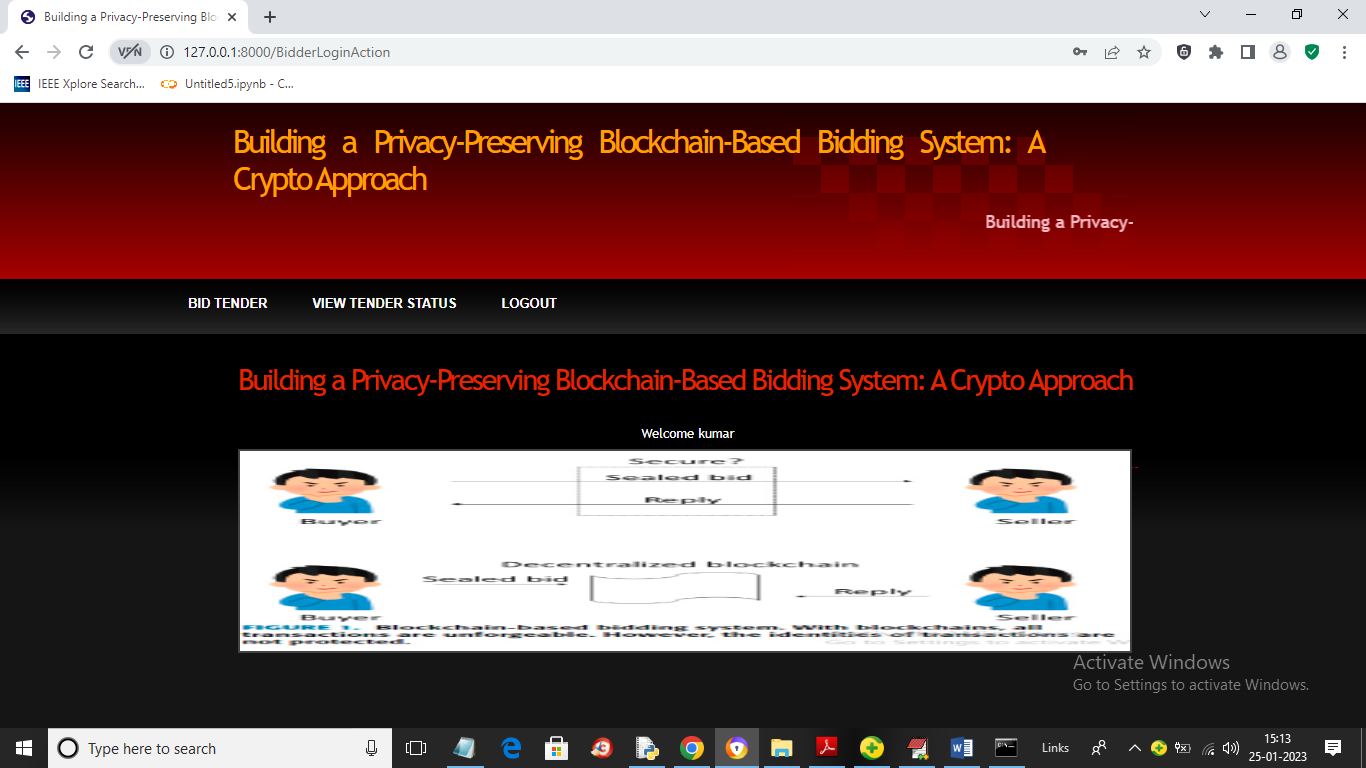
In above screen another user ‘John’ is signing up and press button to get below page



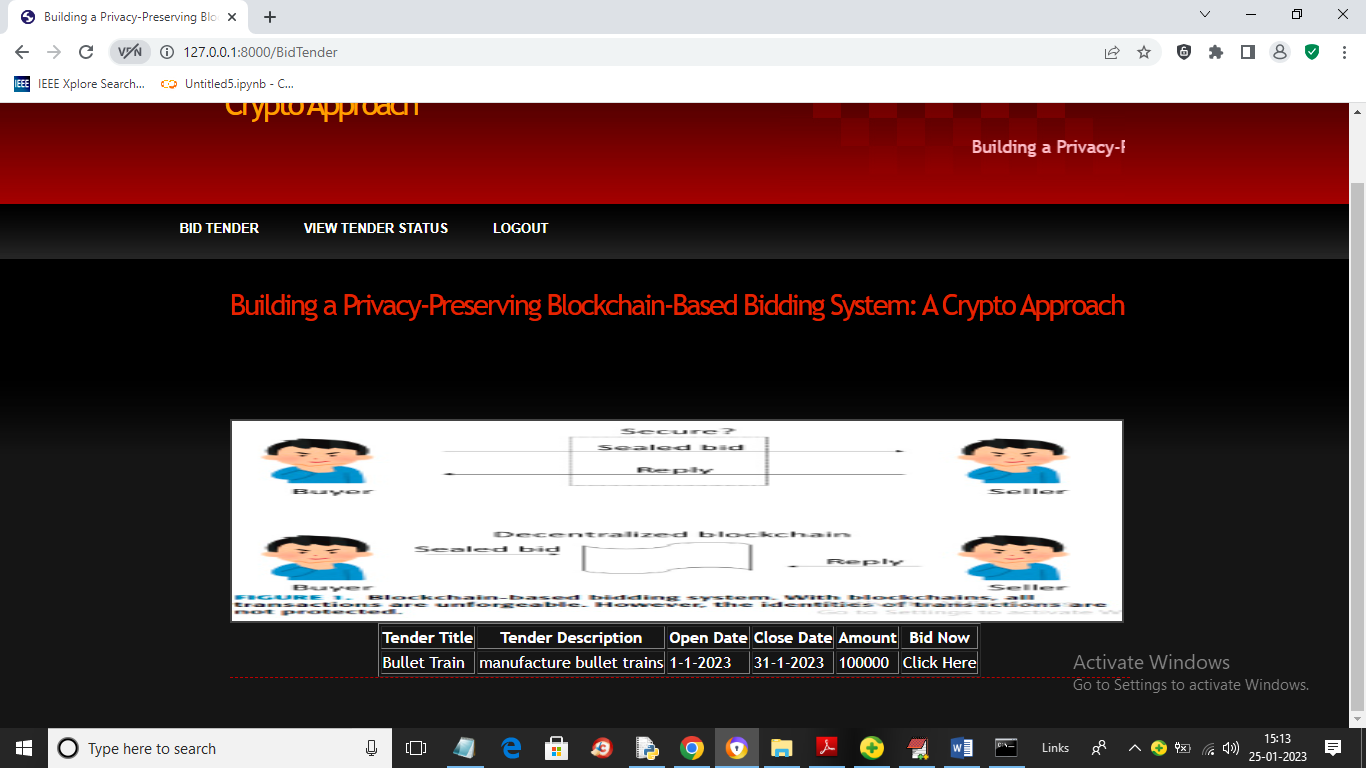
In above screen signup completed and now login as one user



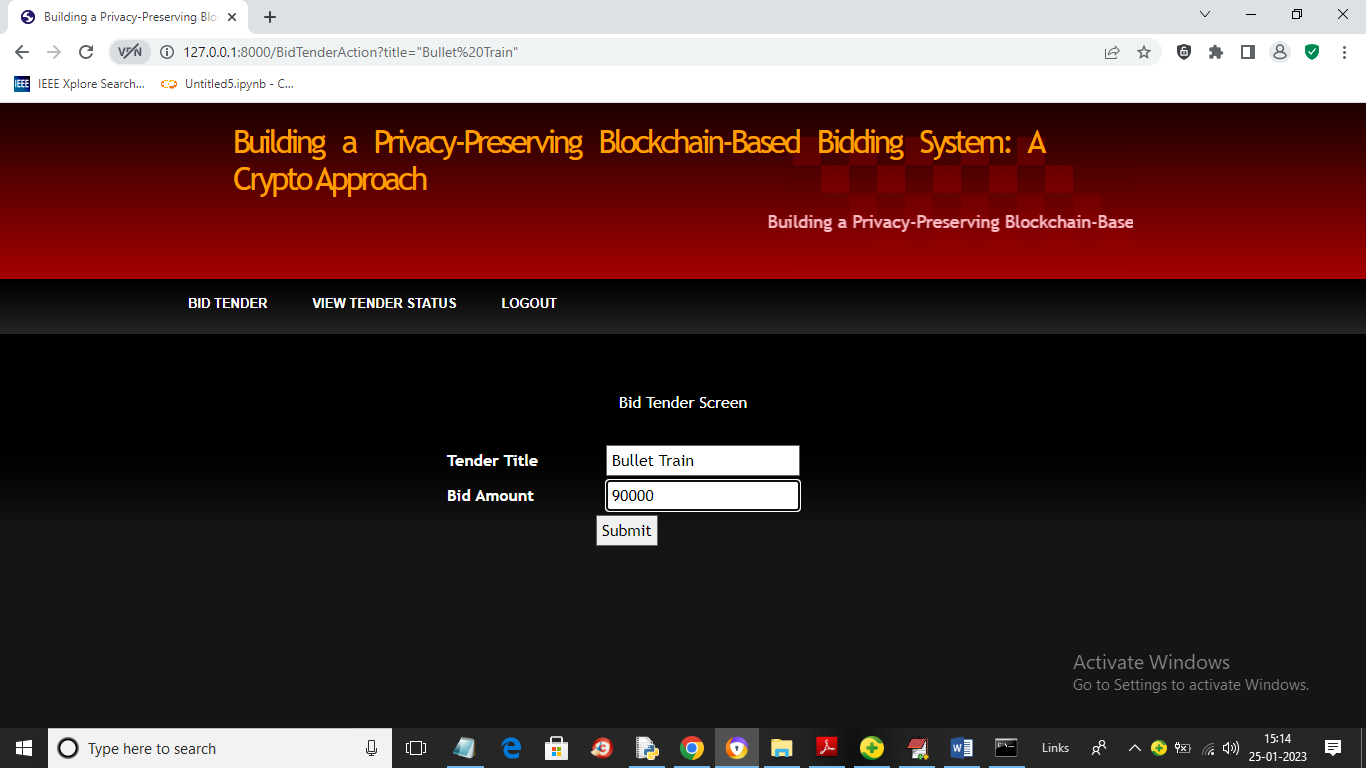
In above screen ‘kumar’ user is login and after login will get below page



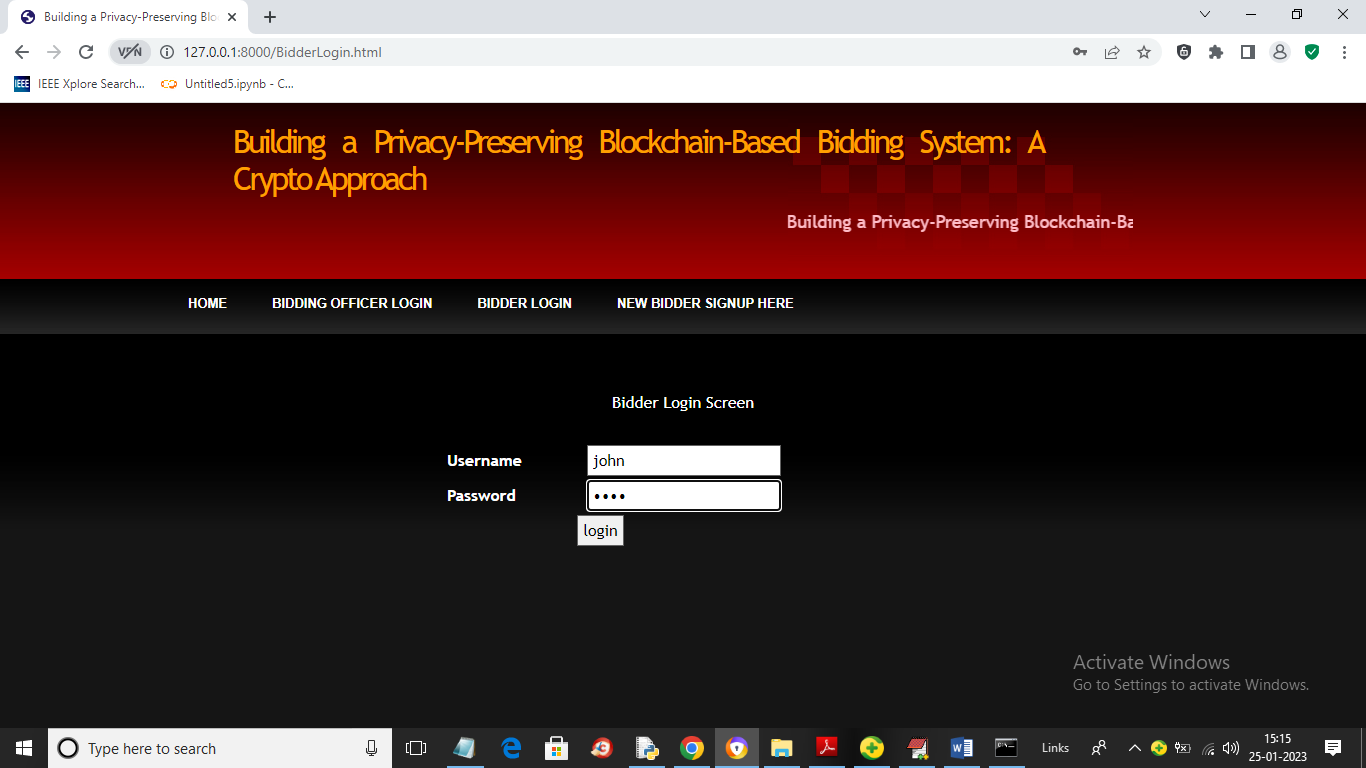
In above screen user can click on ‘Bid Tender’ link to give bidding on available tenders



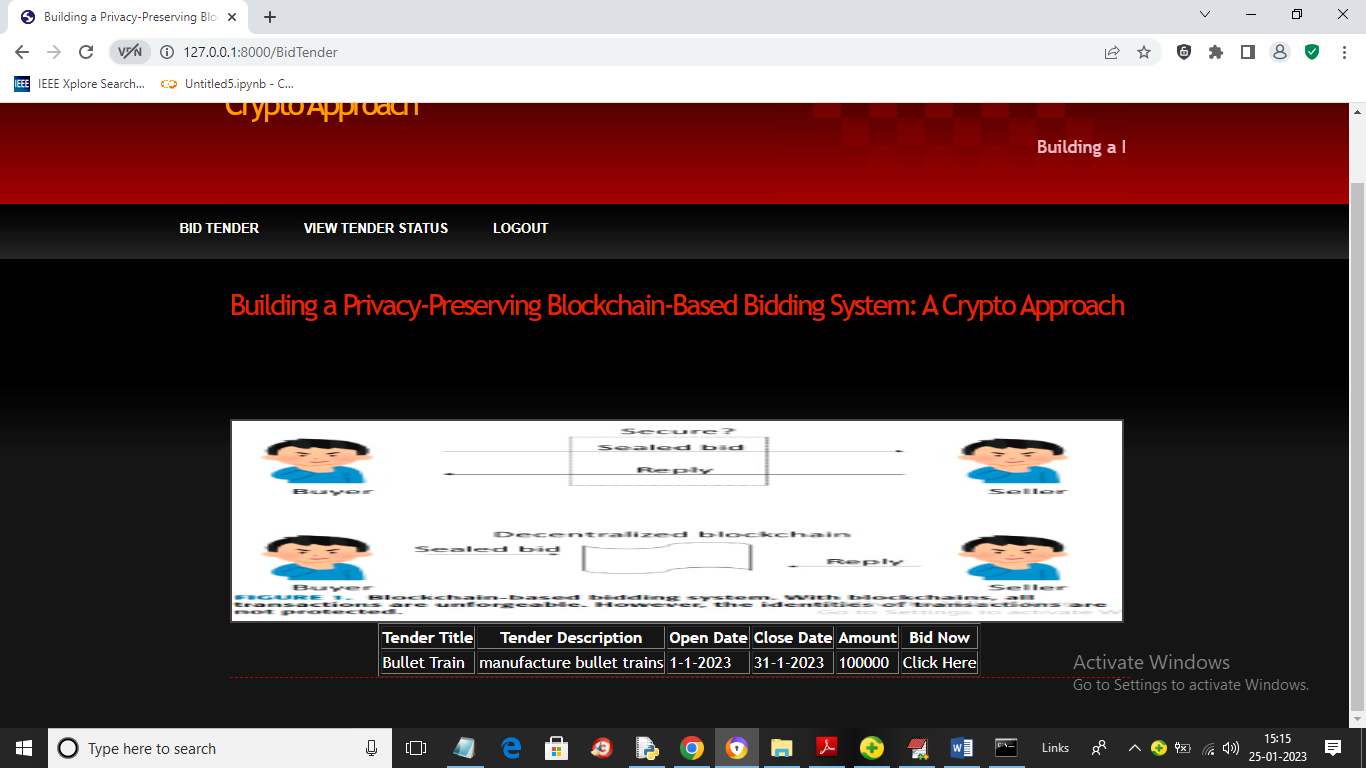
In above screen user can view list of tenders and can click on ‘Click Here’ link to give bids like below screen



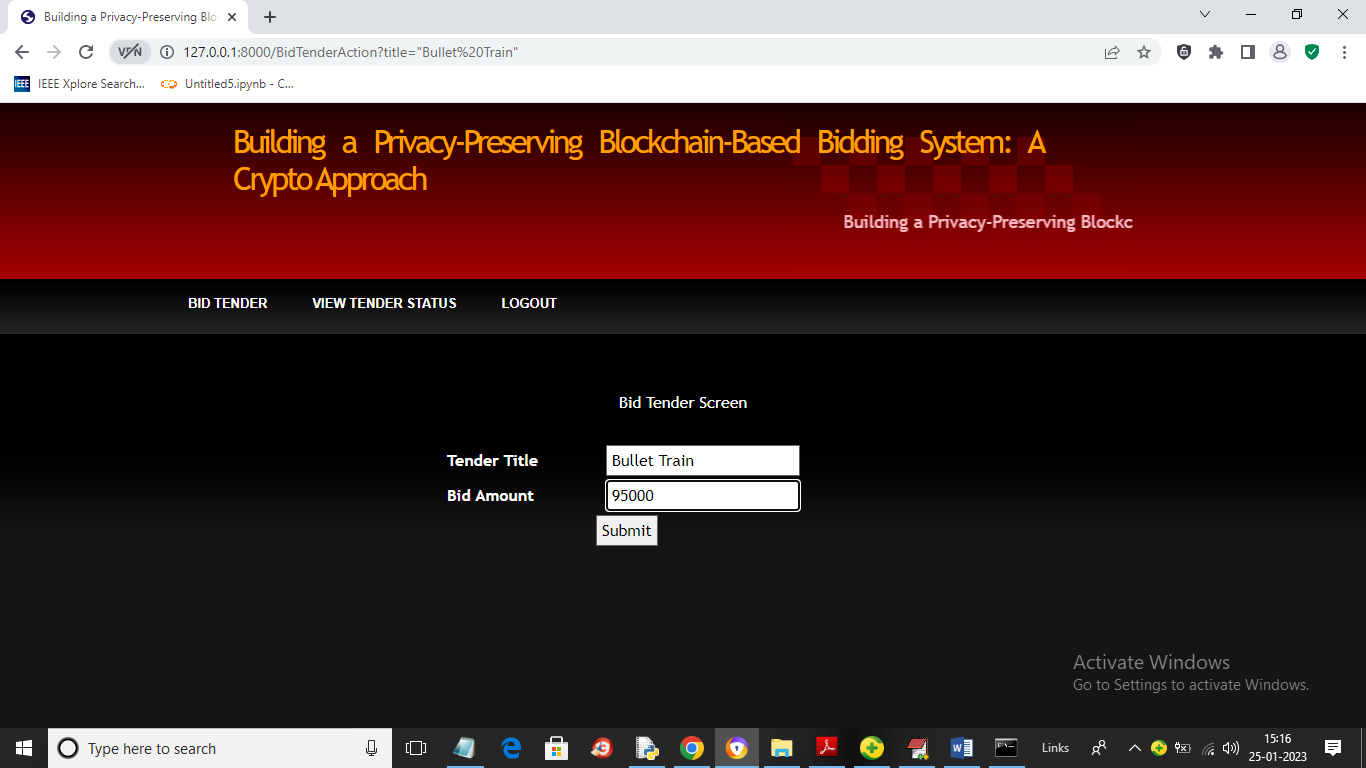
In above screen kumar user giving bidding as 90000 and now login as another user and give bidding



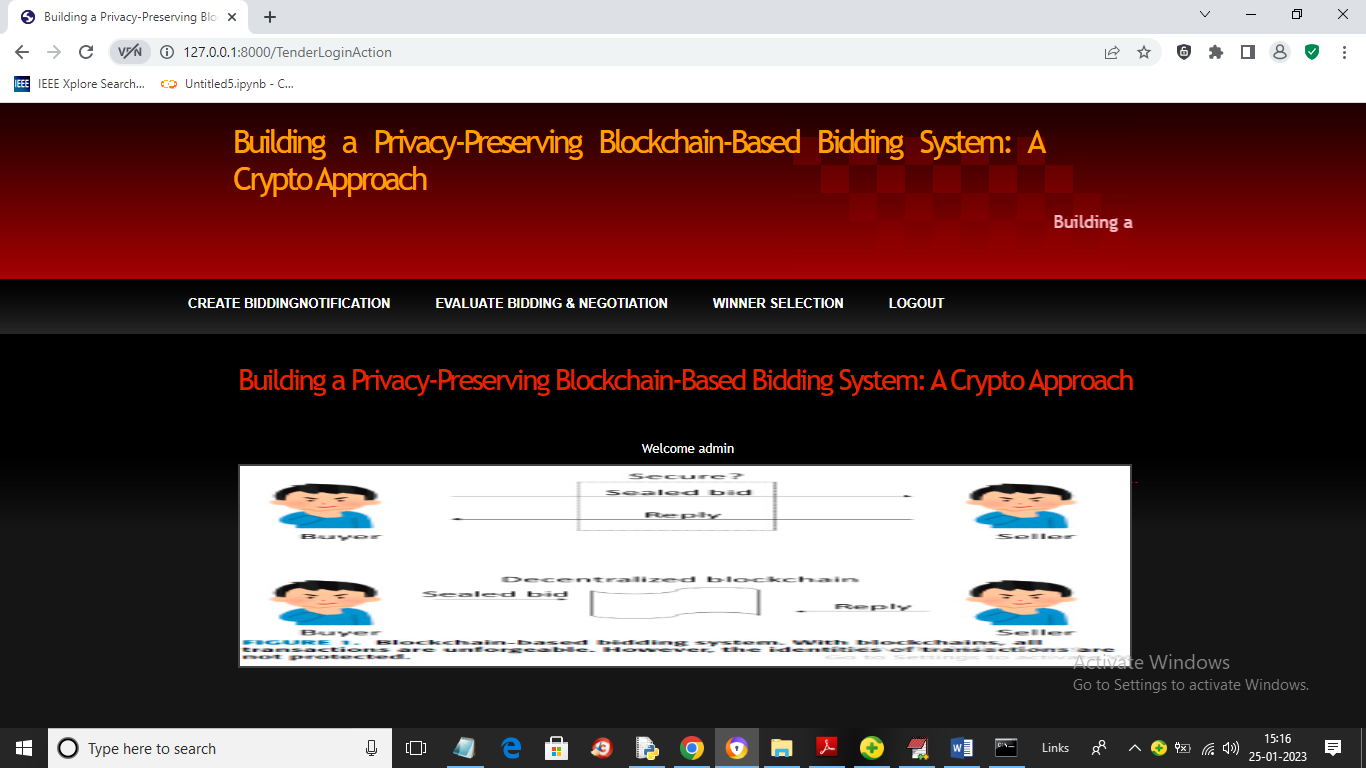
In above screen user John is login and after login will get below page



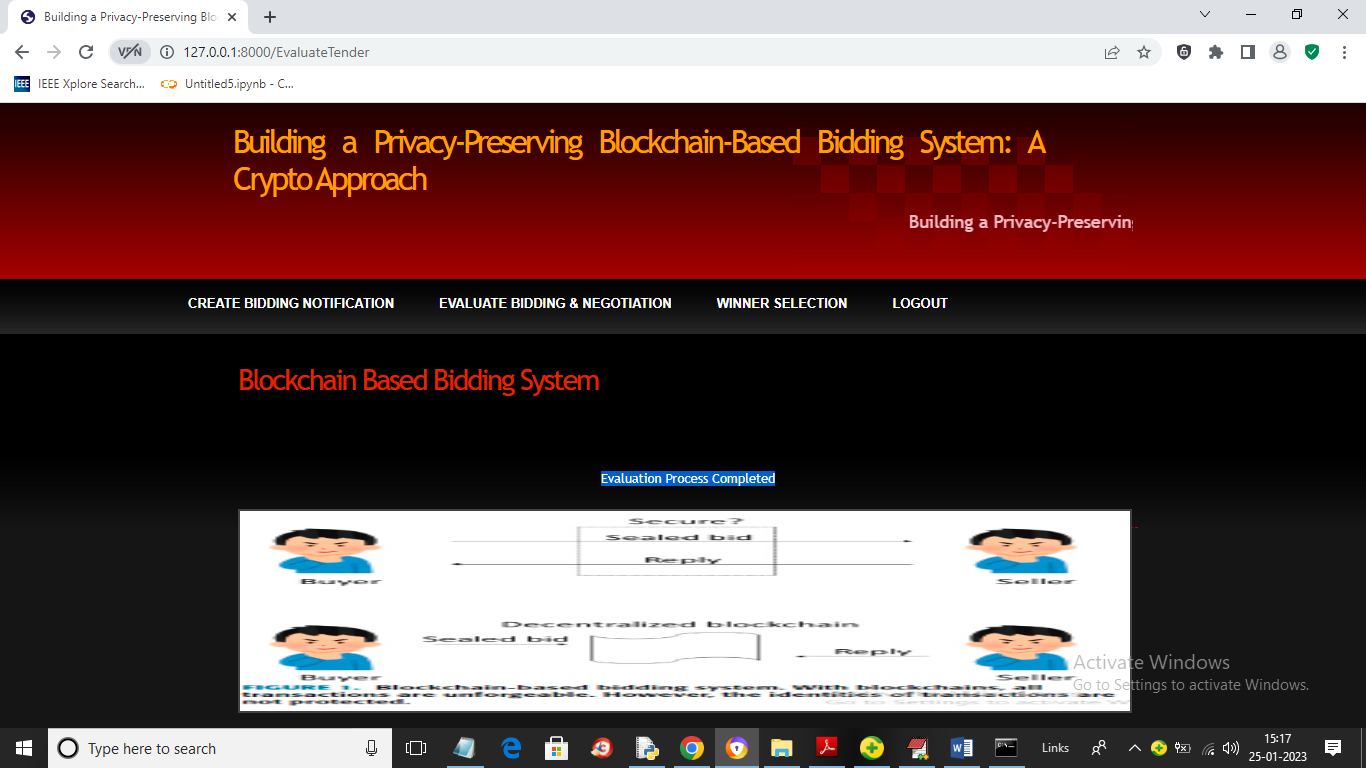
In above screen John user can click on ‘Bid Tender’ link to give bidding



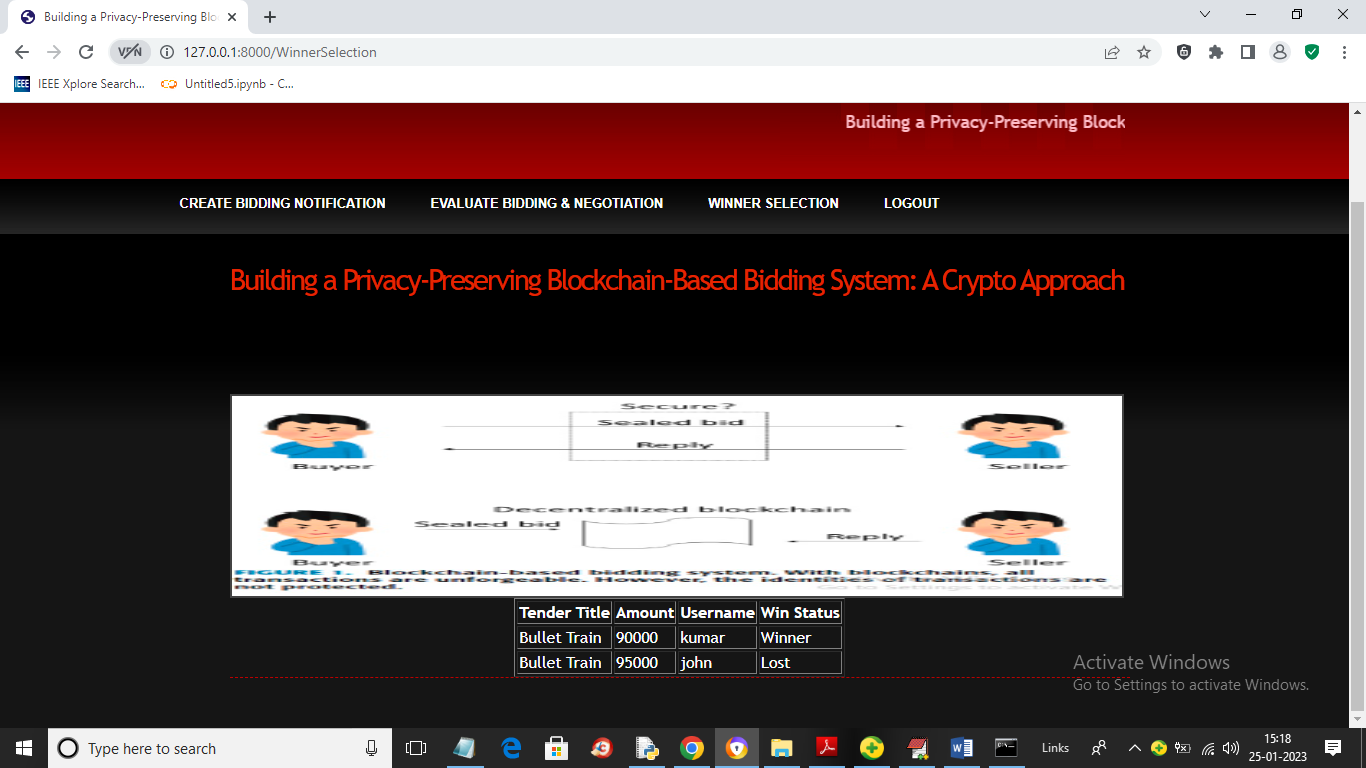
In above screen John is giving 95000 bidding and now logout and login as ‘Bidding officer’ to select winner with low bidding



In above screen bidding officer can click on ‘Evaluate Bidding & Negotiation’ link to select winner like below screen



In above screen evaluation process completed and now click on ‘Winner Selection’ link to view winner like below screen



In above screen we can see kumar and john bid for same tender but kumar gave lowest bid as 90000 so he got selected as winner and similarly bidder can login and view status of winner.

In above screens we can see all users who bid for same tender can decrypt and view bidding and winner details.