A

PROJECT REPORT

ON

**ONLINE BIDDING SYSTEM**

B.Tech (CE) Sem-VI

**In fulfilment of all requirements for**

**Bachelor of Technology**

**In**

**Computer Engineering**

**SEM VI**

**In the Subject of**

**System Design Practices**

**Ravinder Singh (CE-98) ( 14CEUOS059)**

Under the Guidance of

**Prof. Nainesh Patel**

****

**DEPARTMENT OF COMPUTER ENGINEERING**

**FACULTY OF TECHNOLOGY,**

**DHARMSINH DESAI UNIVERSITY**

COLLEGE ROAD, NADIAD- 387001

**DHARMSINH DESAI UNIVERSITY NADIAD-387001,**

**GUJARAT**

****

**CERTIFICATE**

**This is to certify that the project carried out in the subject of**

**System Design Practices entitled “ONLINE BIDDING SYSTEM” and recorded in this report is a work of**

|  |  |  |
| --- | --- | --- |
| **Ravinder Singh** | **ROLL NO: CE-98** | **ID:15CEUOS059** |

**of Department of Computer Engineering, semester VI. They were involved in Project developing during academic year 2016 -2017.**

Prof. Nainesh Patel Dr. C.K.Bhensdadia,

(Project Guide), Head,CE Dept.,

CE Dept,. Faculty of Technology,

Faculty of Technology Dharmsinh Desai University,Nadiad

Dharmsinh Desai University, Nadiad

**CONTENTS**

**Abstract** **i**

[**1. Introduction**](#page6) **1**

[**1.1 Project details: Broad specifications**](#page6) **2**

[**1.2 Technology Used**](#page6) **2**

**2.** [**Software Requirement Specifications**](#page7) **4**

[**2.1 Purpose**](#page7) **5**

[**2.2 Scope**](#page7) **5**

[**2.3 O**](#page7)**verall Description 5**

[**2.3.1 P**](#page8)**roduct Perspective 5**

[**2.3.2 P**](#page8)**roduct Functions 5**

[**2.3.3 User Classes and Characteristics**](#page8) **6**

[**2.3.4 Operating Requirements**](#page8) **6**

[**2.3.5 D**](#page9)**esign and Implementation Constraints 6**

[**2.3.6**](#page9) **Assumptions and Dependencies 7**

[**2.4**](#page9) **External Interface Requirements 7**

[**2.4.1**](#page9) **User Interfaces 7**

[**2.4.2**](#page9) **Hardware Interfaces 7**

[**2.4.3**](#page9) **Software Interfaces 8**

[**2.4.4**](#page9) **Communication Interfaces 8**

[**2.5**](#page9) **Funtional Requirements 8**

[**3. Design**](#page10) **11**

[**3.1 Use-case Diagram**](#page10) **12**

[**3.2 Class Diagram**](#page10) **13**

[**3.3 Sequence Diagrams**](#page11) **13**

[**3.4 Activity Diagrams**](#page12) **14**

[**3.5 State Diagram**](#page12) **15**

[**3.6 E**](#page13)**-R Diagram 16**

[**4. Implementation**](#page13) **17**

[**4.1 Implementation Environment**](#page13) **18**

[**4.2 Modules Description**](#page13) **18**

[**5. Testing**](#page14) **21**

[**5.1 Testing Plan**](#page14) **22**

[**5.2 Testing**](#page14) **Strategy 22**

[**5.3 Testing**](#page14) **Methods 22**

[**5.4 Test**](#page14) **Cases 23**

[**6.**](#page14) **Screenshots 25**

[**7.**](#page14) **Limitations and Future Enhancements 31**

[**8.**](#page14) **Conclusion 33**

[**9.**](#page14) **Bibliography 35**

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| **3.1** | **Use-Case Diagram** | **12** |
| **3.2** | **Class Diagram** | **13** |
| **3.3** | **Sequence Diagram** | **13** |
| **3.4** | **Activity Diagram** | **14** |
| **3.5.1** | **State Diagram of**  **Start Auction** | **15** |
| **3.5.2** | **State Diagram of**  **Place Bid** | **15** |
| **3.6** | **E-R Diagram** | **16** |
| **6.1.1** | **Register Page** | **26** |
| **6.1.2** | **Log-In Page** | **26** |
| **6.2** | **Home Page** | **27** |
| **6.3** | **Your Products Page** | **27** |
| **6.4** | **Admin Panel Page** | **28** |
| **6.5** | **Auction Page** | **28** |
| **6.6.1** | **T&C for adding products** | **29** |
| **6.6.2** | **Add Products Page** | **29** |
| **6.7** | **Auction Closed Page** | **30** |
| **6.8** | **Product Submission Page** | **30** |

**Abstract**

Products are mainly bought brand new and when they become obsolete for us , they are cornered out . Thus , auctions came into picture where people can earn some money by selling their products or antics to collectors or the ones who need to buy products for themselves.

In the era of digitization , Online Auctions can be more of a use instead of those conducted in a closed room as auctioneers get a bigger scope of products virtually and helps them to attend auctions from their home itself. Keeping that advantage in mind , this project aims to fulfill the demand of conducting such auctions where in auctioneers are not bound by physical presence ,instead , they can participate in the auction making the process of auctioning much more interesting.

**CHAPTER : 1**

**INTRODUCTION**

1. **Introduction**

**1.1** **Project Details: Broad Specifications**

Online Bidding System is a web application . The main aim behind developing this system is to ease out the process of bidding by conducting Auctions online . The web application thrives to provide real-time auctioning with the advantage for the auctioneers to take part in the auction from any part of the world effectively.

This system allows the user to list his/her own product in the auction , which allows the user to take part actively in the auction .

**1.2** **Technology Used**

**Front End : HTML , CSS , AngularJS**

As the system is a web application , the need of HTML and CSS is understood . Here , one more technology for the modeling of data ,that is , AngularJS.

AngularJs is a Javascript Framework. It can be added to a HTML page within the script. It extends HTML attributes with Directives and binds data to HTML with Expressions.

**Back End : Node.JS**

The project is built using mean stack tools , where the back end code in written in Node.JS which is , again a Javascript Framework. The routing is done using ExpressJS

**Database : MongoDB**

MongoDB is an open-source cross-platform document-oriented database program. Classified as NoSQL database program, MongoDB uses JSON-like documents with schemas .( JSON - Javascript Object Notation )

**Diagram Tool : Umlet**

All the UML diagrams and various other diagrams corresponding to this project are made in Umlet.

**CHAPTER : 2**

**SOFTWARE REQUIREMENTS**

**SPECIFICATIONS ( SRS )**

**2. Software Requirement Specifications**

**2.1 Purpose**

This is a SRS document refers to Online Bidding System Release 2017 version 1. It describes the functionality and specification of how the Online Bidding System will help in serving the customers in an efficient way. It also specifies how it will interact with the end users.

**2.2 Scope**

This system is designed to conduct Bidding in an efficient manner where the customers can take part in bids online without having much trouble of being unnoticed.

System provides an interface for conducting auctions online by providing an user- convenient interface.System provides a mechanism for the vendors to present their products for the bidders and a live surveillance for getting updates of the auction.



**2.3 Overall Description**

**2.3.1 Product Perspective**

This software is developed as a part of course work the subject “System Design Practice”. The software aims to conduct bidding / auction online with ease.The web- app ‘ s main perspective is towards time management and provide a user-flexible system.

**2.3.2 Product Functions**

The product offers a light weight User Interface to work with and provides a secure way to deal with the management stuff. The product also aims at making the auction system online and along with the admin panel ,design auctions in a very efficient manner without having much trouble.The product also helps the auctioneers to have a clear view of the ongoing bids and an easy mechanism for placing their own bids for their desired products.

**2.3.3** **User Classes and Characteristics**

Basically there are two types of end users i.e.,

(1) Customers:-

System is developed for people of all genre and mainly for those who are actively involved in going to various auctions .

(2) Vendors:-

The ones who come to the auction to sell their products ie. Place them in auctions .

**2.3.4 Operating Environment**

The system is a web-app and not an android or ios application because development in different operating environment will take time and cross-platform development is not the base of the development team and a web-app can be used by everyone in an efficient way.

**2.3.5 Design and Implementation Constraints**

The constraints basically are financial as well as at a corporate level . The system aims to unite all the vendors and the customers in one network for conducting an online auction so their is a need for the vendors to place their products and the customers taking part in the auction .So , public baking is a constraint for this system and at a financial level ,this product requires web hosting on a very secure server with a very good speed which is costly as well as digital media publicity is required for making this web-app well known enough to conduct the auction .

**2.3.6 Assumptions and Dependencies**

System may not work if appropriate network is not available.Mainline network of system highly needs electricity to work on.System requires the customers to be online at a particular time for starting the auction or else the bids are missed.System requires the vendors to provide the product details before a particular stipulated time of the auction else he/she cannot take part in the bidding .

**2.4** **External Interface Requirements**

**2.4.1 User Interfaces**

Application can be accessed through any browser interface. The software will be fairly compatible with Microsoft Internet Explorer Version 6 and above or other modern web browsers.

**2.4.2 Hardware Interfaces**

**Server Side**:

Operating System: Linux.

Processor: Pentium 3.0 GHz or higher.

RAM: 2GB or higher.

Hard-Disk: 100GB or more.

**Client Side:**

Operating System: Windows 7,8, 8.1,10.

Processor: Pentium 3.0 GHz or higher.

RAM: 2GB or higher.

**2.4.3 Software Interfaces**

**Client Side:**

HTML supported Web Browsers, Windows 7,8, 8.1, 10 ,MAC OS, Linux (All Flavors).

**Server Side:**

Linux , MongoDB Database.

**2.4.4 Communication Interfaces**

The Online Bidding System shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

**2.5** **Functional Requirements**

**R1**: System provides a feature for customers and vendors to create their accounts for taking part in auction

**Input** : Customer Details

**Output** : Account Creation Status

**Processing** : The details added by the customer are stored into the database and the customer is notified for the same .

**R2**: System requires the vendors and customers to provide the details for logging in to the system for being active in the auction.

**Input** : Customer / Vendor Credentials

**Output** : Home Page.

**Processing**: System takes the user credentials and compares them with the database and provides appropriate output.

**R3**: System provides an interface for the customers to place their bids for the products that are displayed.

**R3.1**: System displays the list of products for the ongoing auction for the bidding for 5 minutes.

**R3.2**: System requires the Customers to place their bids parallely for a desired product and the system selects a winner out of all the auctioneers who has placed the highest bid

**R4**: System allows vendors to submit the list of products to be placed for the auction .

**R4.1**: System stores the information of various products enlisted by the vendors.

**Input** : Product details and starting bid

**Output** : Stored into the database.

**Processing** : System stores the information in the database accordingly.

**R5**: System provides an admin panel

**R5.1**: System provides the maintenance of products

**R5.2**: System keeps a track of highest bidders for particular products

**R5.3**: System provides surveillance over customers details and vendor details.

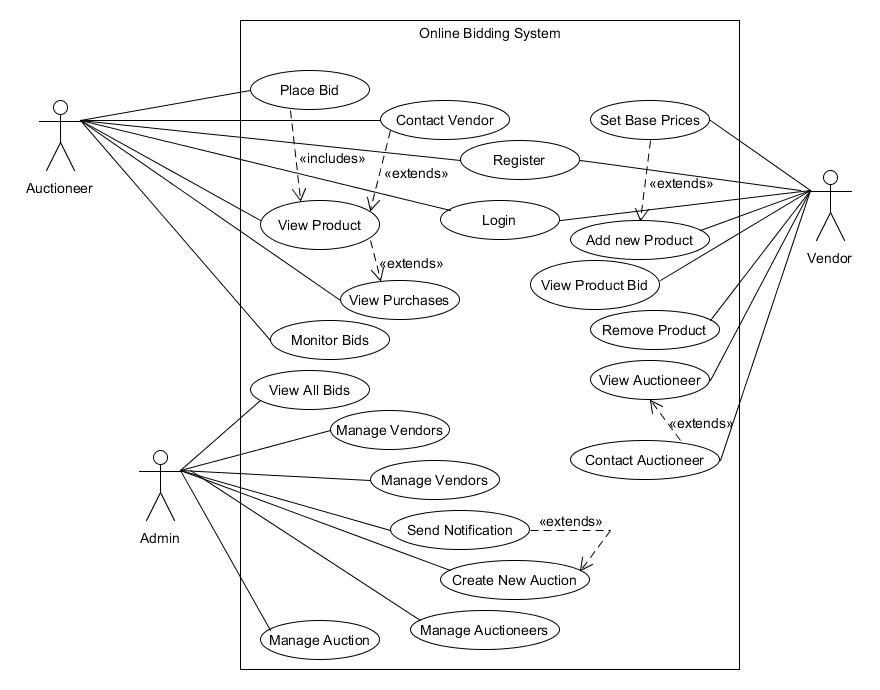
**R6**: System provides a live surveillance for the customers during the auction to keep a track of ongoing bids.

**CHAPTER : 3**

**DESIGN**

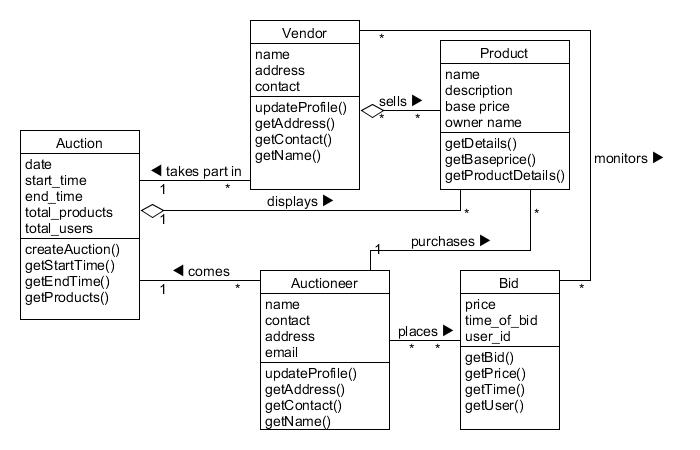
**3. Design**

**3.1 Use-Case Diagram**

****

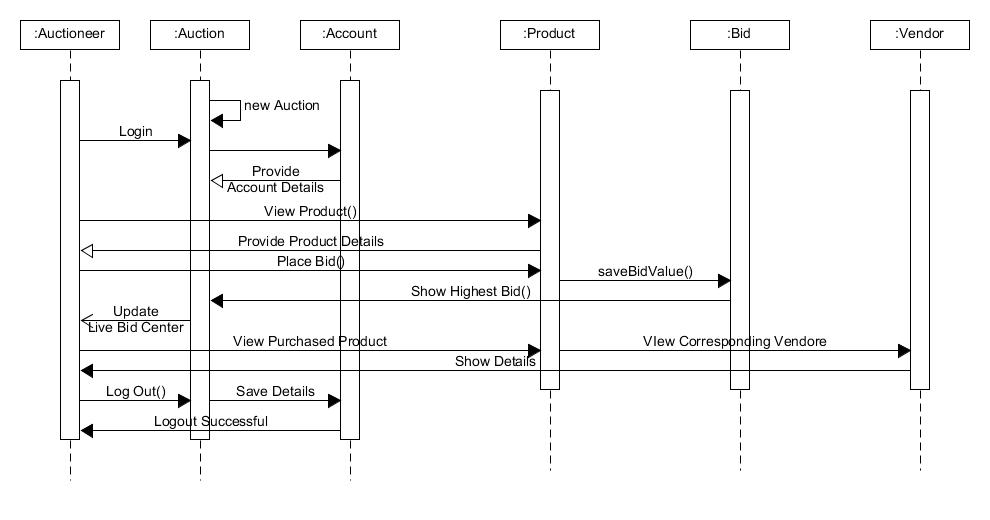
**Figure 3.1 Use Case Diagram for Online Bidding System**

**3.2 Class Diagram**

****

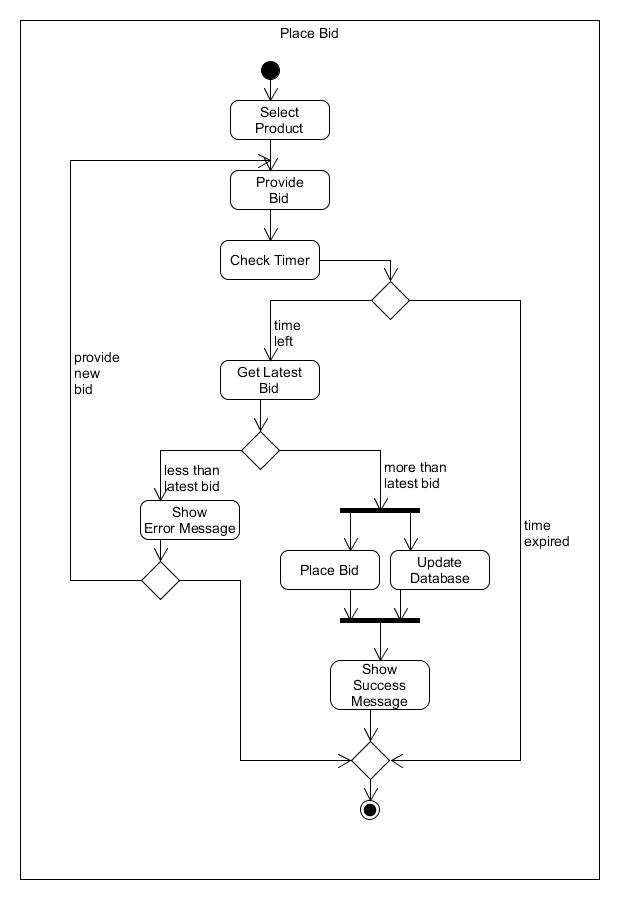
**Figure 3.2 Class Diagram for Online Bidding System**

**3.3 Sequence Diagram**

****

**Figure 3.3 Sequence Diagram for Online Bidding System**

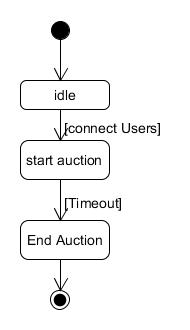
**3.4 Activity Diagram**

****

**Figure 3.4 Activity Diagram for Online Bidding System**

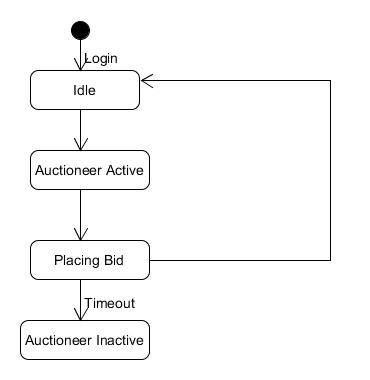
**3.5 State Diagrams**

**3.5.1 Start Auction State Diagram**

****

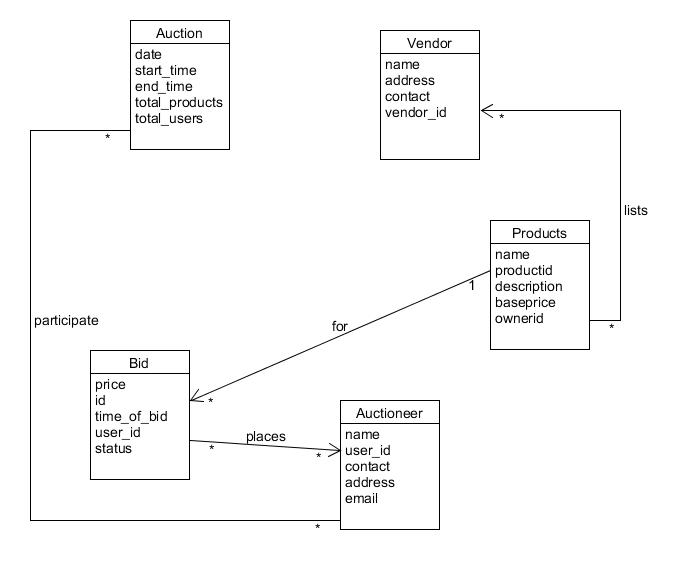
**Figure 3.5.1 State Diagram of Start Auction for Online Bidding System**

**3.5.2 Place Bid State Diagram**

****

**Figure 3.5.2 State Diagram of Place Bid for Online Bidding System**

**3.6 E-R Diagram**

****

**Figure 3.6 E-R Diagram for Online Bidding System**

**CHAPTER : 4**

**IMPLEMENTATION**

**4. Implementation**

**4.1** **Implementation Environment**

* Node.JS environment in Windows 10
* MongoDB Client in Windows 10

**4.2 Modules Description**

* **Register / Log - In Module**

It is used to store information of users who are new to the system and accessing the system for the first time and also authenticating the users before they can explore various auctions listed in the system.

**Input** : User’s information or credentials

**Output**:Stored or Verified successfully

**Processing**: Check user’s credentials in the database while logging in or store them in the database while registering new user.

* **Store Product Module**

This module comes into action when a user lists his/her product for any particular auction . It considers all the details of the respective product and stores its details in the MongoDB client and the respective product’s image in the local folder.

**Input** : Product’s Information

**Output**:Stored successfully

**Processing**: Product’s information is stored in the database and corresponding image in the local folder ( at server end ).

* **Live-Auction Module**

This is the main module for the system. It involves the functionality for conducting real-time auction.

Here, **socket.io** is used for performing real-time event-driven bidding experience . It comprises of 5 events:

1. New Auctioneer Event

Triggers whenever a new auctioneer joins the auction.

1. Send Product and Timer Event

After new auctioneer arrives or timeout occurs ,this particular event will be triggered for sending the listed product for that auction and time left for bidding for that particular product.

1. Timeout Event  
     
   Whenever the timer goes off,this event is triggers to update the variables set for every auctioneer using their socket-id and new product is sent by triggering send product event.
2. Auctioneer left Event  
     
   Whenever an auctioneer leaves the auction , corresponding space is freed and all users are notified of it by triggering this event.
3. Auction Finish Event

This event is triggered whenever the auction gets over providing the over-all information of products sold and their sell-price and other corresponding information.

* **Admin Control Module**

This module is used by the administrator of the system in order to maintain control of the present auctions or to create auctions. Here ,all the products listed by the users come for verification .Thus ,only after the admin approves the product ,it can be listed for the auction.

**CHAPTER : 5**

**TESTING**

1. **Testing**

**5.1 Testing Plan**

The testing is a technique that is going to be used in the project is black box testing ,the expected inputs to the system are applied and only the outputs are checked .

**5.2 Testing Strategy**

The development process repeats this testing sub process a number of the lines for the following phases.

* Unit Testing
* Integration Testing

Unit Testing tests a unit of code after coding of that unit is completed. Integration Testing tests whether the previous programs that make up a system, interface with each other as desired. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification.

Testing is carried out in such a hierarchical manner to that each component is correct and the assembly/combination of component is correct. Merely testing a whole system at end would most likely throw up errors in component that would be very costly to trace and fix. We have performed both Unit Testing and System Testing to detect and fix errors.

**5.3 Testing Methods**

We have performed Black-box testing for the testing purpose. A brief description is given below:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

**5.4 Test Cases**

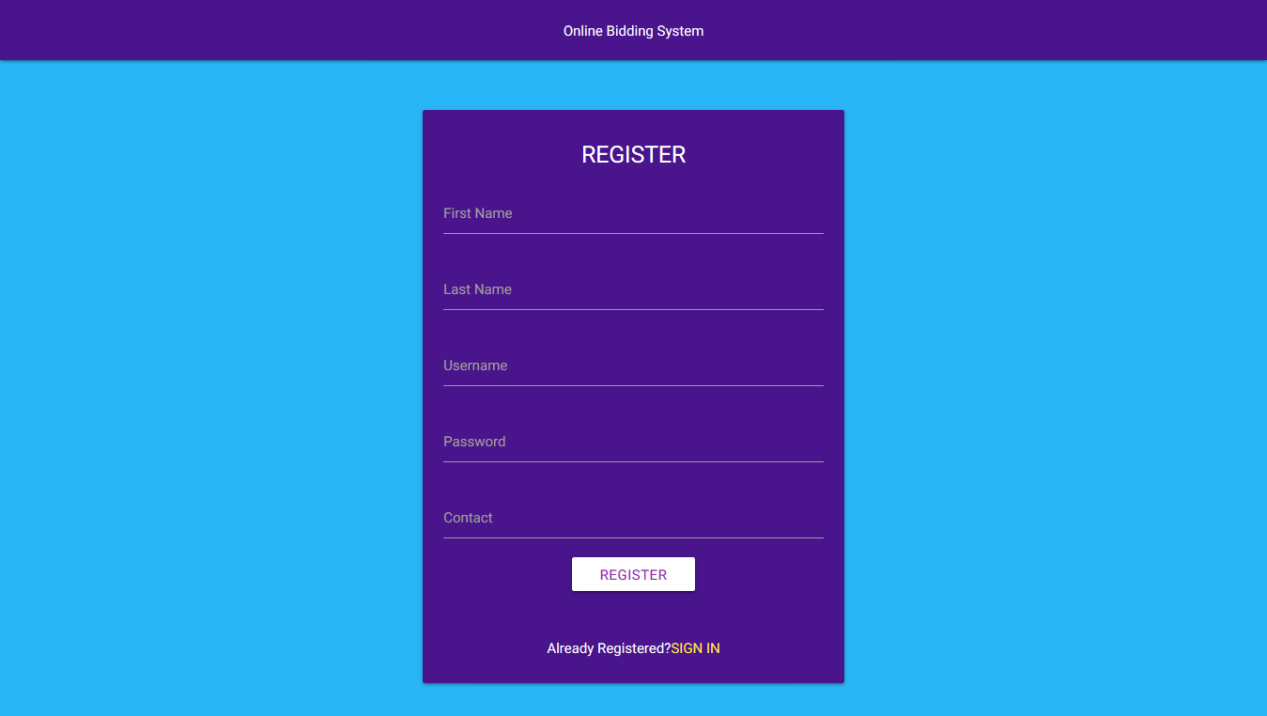
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test**  **Case ID** | **Test**  **Scenario** | **Test**  **Steps** | **Test**  **Data** | **Expected**  **Results** | **Actual**  **Results** |
| T01 | Sign Up | 1. Go to Homepage 2. Provide   Information | User  Information | Stored  Successfully | Success |
| T02 | Log In | 1. Go to Homepage 2. Enter Credentials | User  Credentials | Display Main Page | Main Page  Displayed |
| T03 | Add Product | 1. Go to Main Page 2. Go to List your Product 3. Enter Product Details | Product  Information | Image Stored on Server and information in database | Success |
| T04 | Join Auction | 1.Check time of the auction | Auction Details | Not Started Page if auction not started else Auction Main page | Result Shown accordingly |
| T05 | New Bid | 1. Go to Auction Main Page 2. Provide New Bid | Bid Value | If less than current Bid then no action else save Bid. | Success |
| T06 | Product Timeout | 1. Go to Auction Main Page 2. Wait for the product to get expired | Product Timer | If timer gets expired ,show new product and save the final bid | Successfully stored bid and sent new Product |
| T07 | Start  /Stop  Auction | 1. Go to Admin Panel 2. Start Auction | Auction Id | Auction Started | Auction Started |
| T08 | Create Tables for Auction | 1. Go to Admin Panel 2. Click Create Tables Button | Auction id and respective products | Table Created | Created  Successfully |
| T09 | Create New Auction | 1. Go to Admin Panel 2. Click New Auction button | New Auction Details | Save the details in the database | Saved Successfully |
| T10 | Verify product | 1. Go to Admin Panel 2. Go to Verify Products Section | Product Information | Verified Successfully | Success |
| T11 | Enable  /Disable Auction | 1. Go to   Admin Pane;  2. Enable Auction | Auction Id | Enabled or Disabled Successfully | Success |

**CHAPTER : 6**

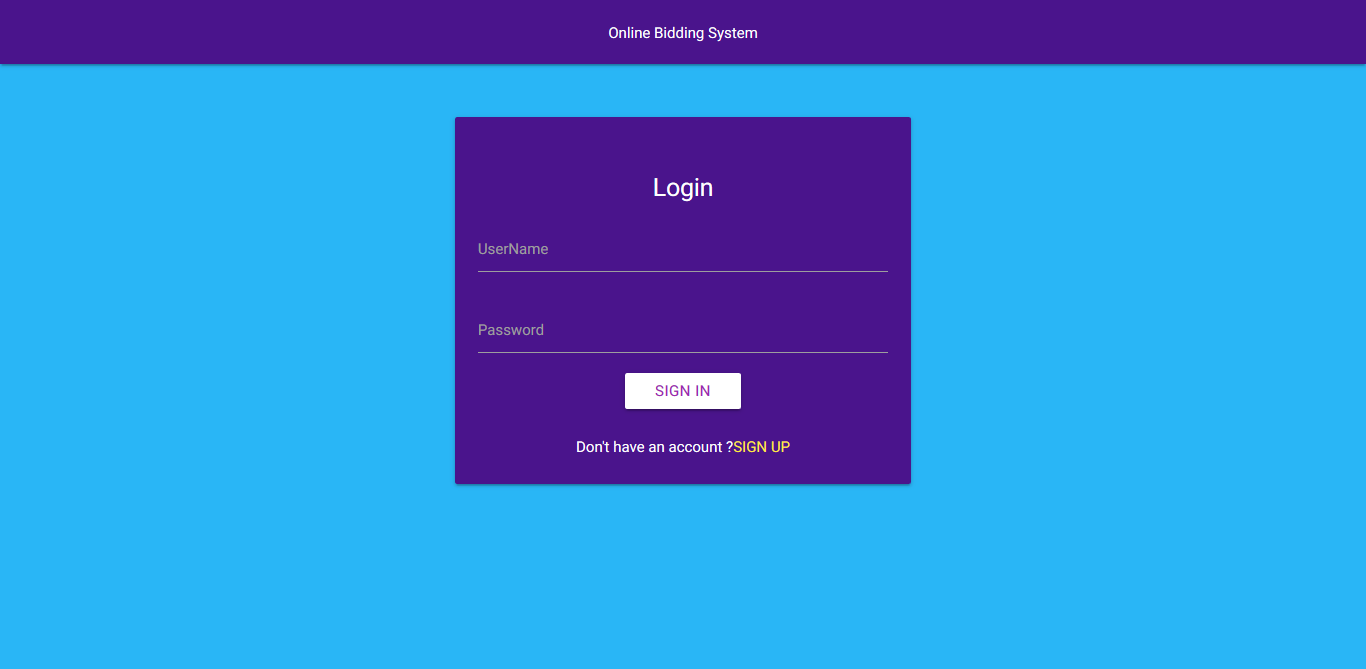
**SCREENSHOTS**

**6. Screen-shots**

**6.1 Register and Log In**

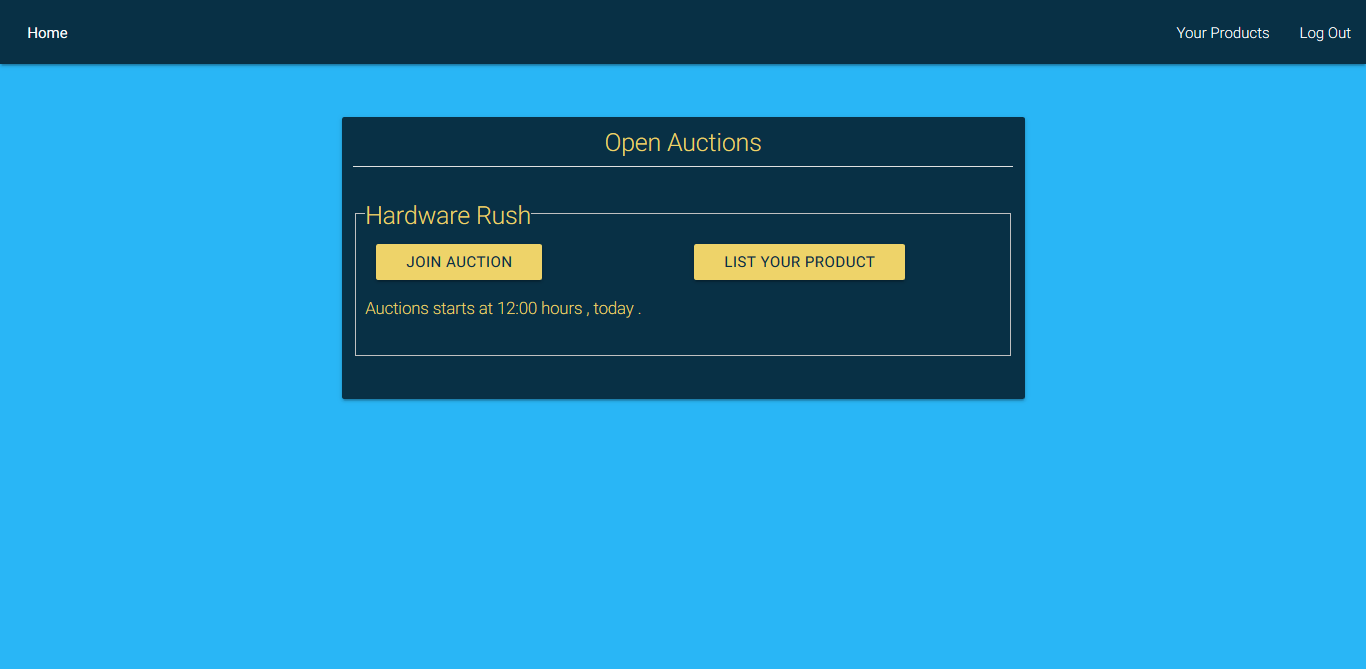


**Figure 6.1.1 Register Page**



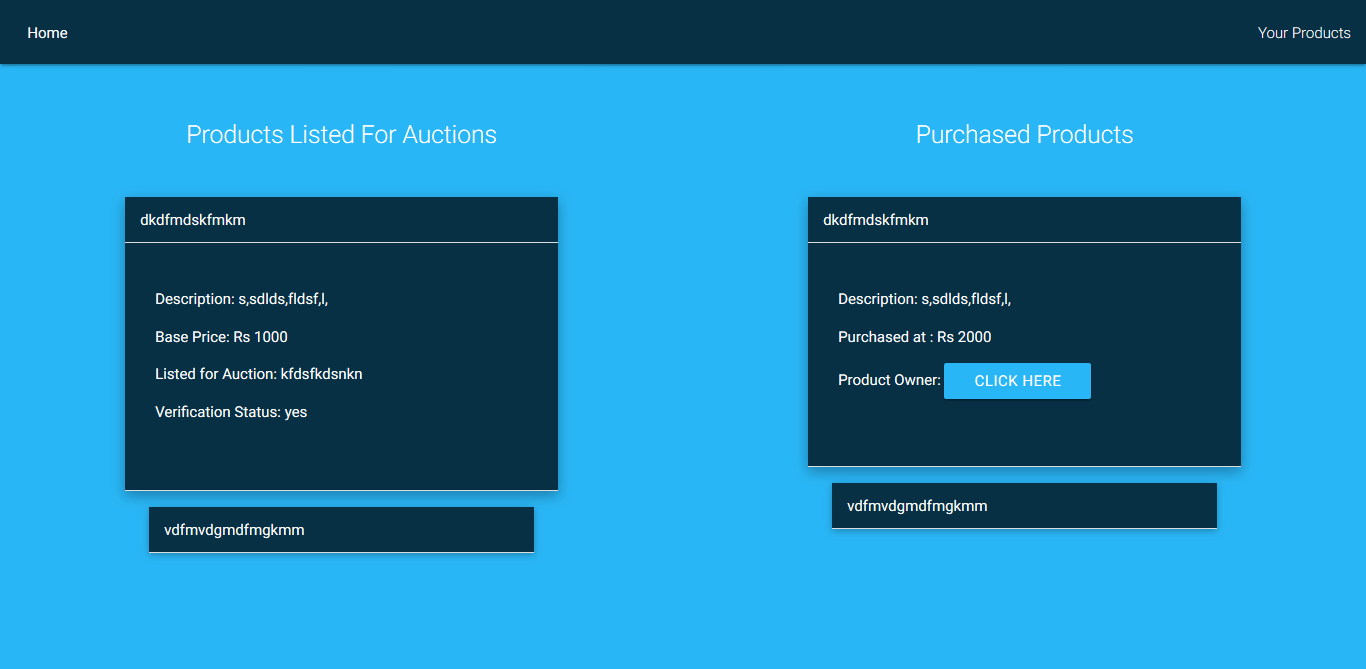
**Figure 6.1.2 Log-In Page**

**6.2 Home**

****

**Figure 6.2 Home Page**

**6.3 Your Products**

****

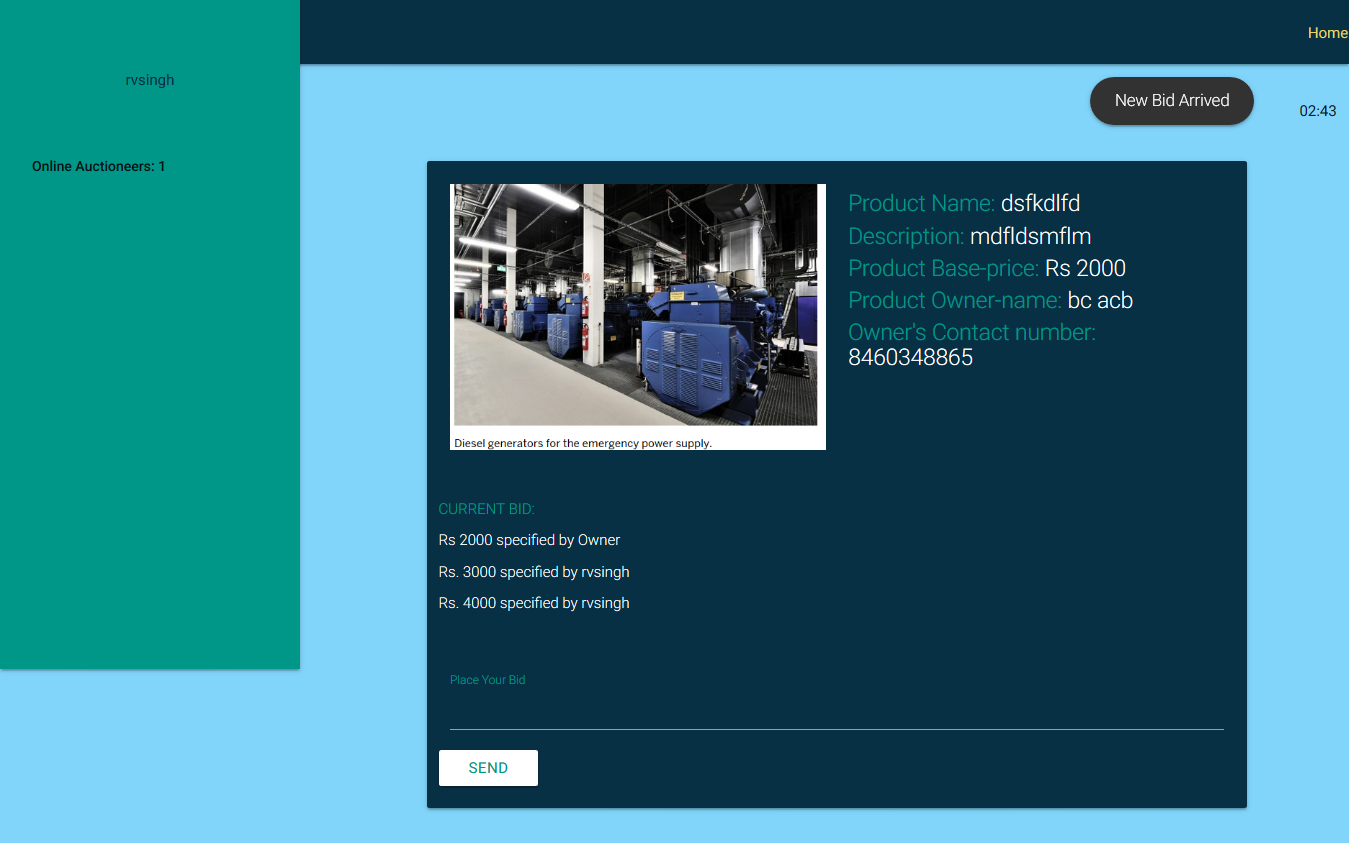
**Figure 6.3 Your Products Page**

**6.4 Admin Panel**

****

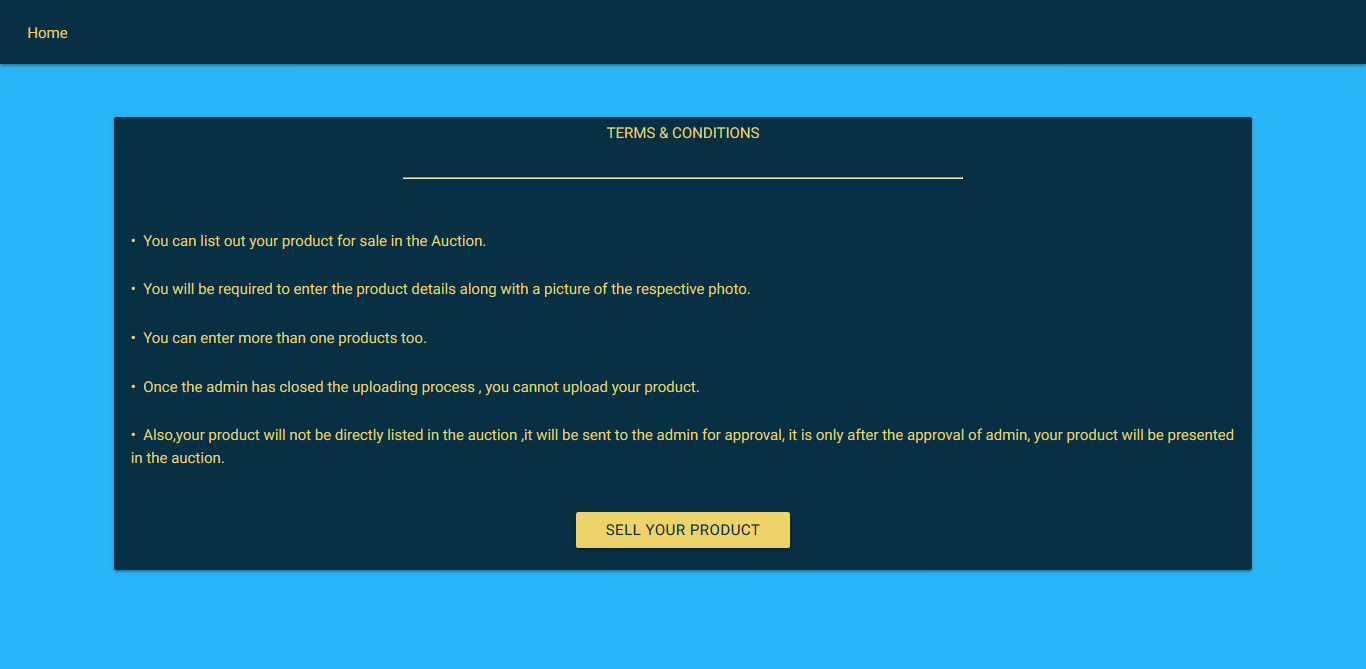
**Figure 6.4 Admin Panel Page**

**6.5 Real-Time Auction**

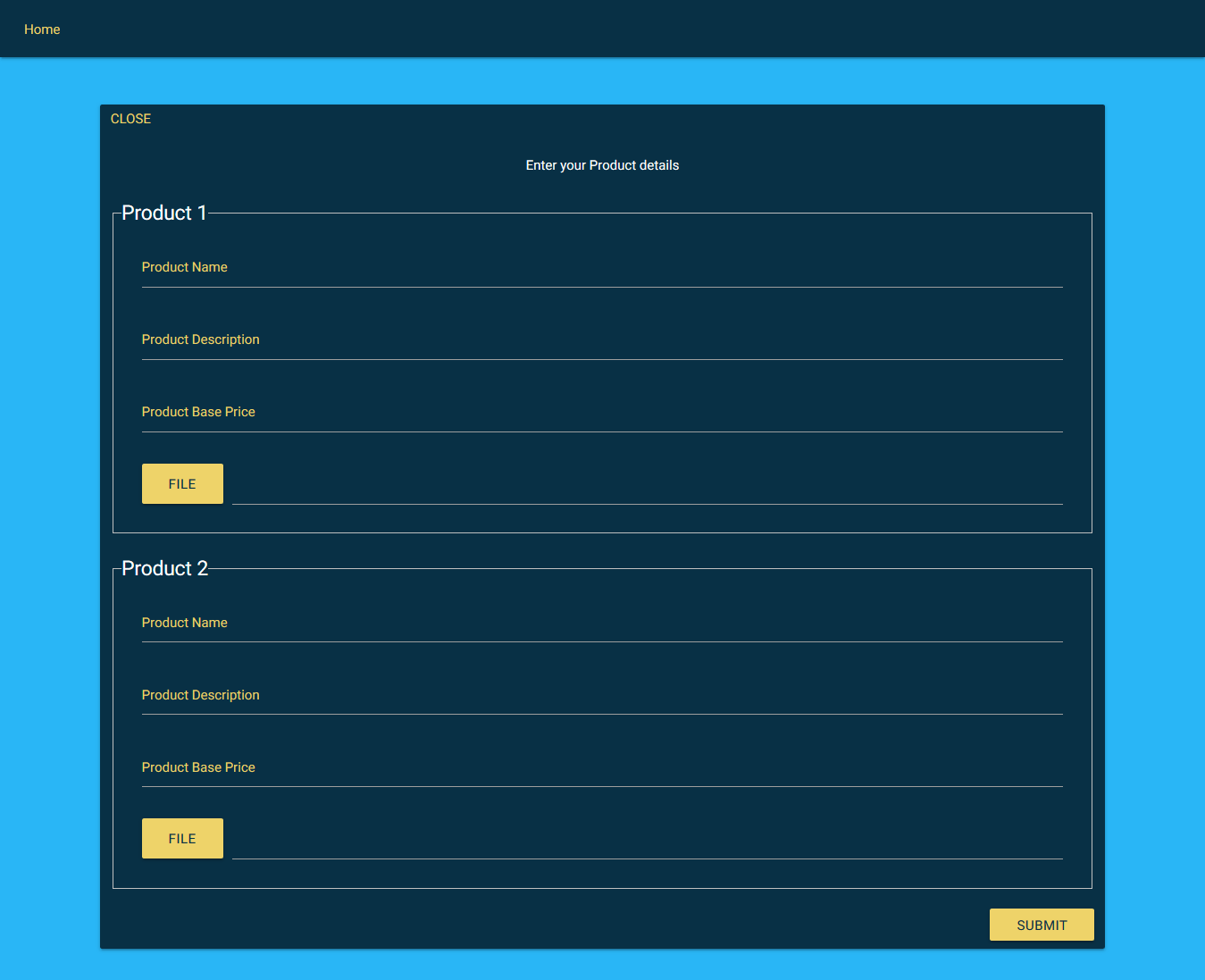
****

**Figure 6.5 Auction Page**

**6.6 Add Product**

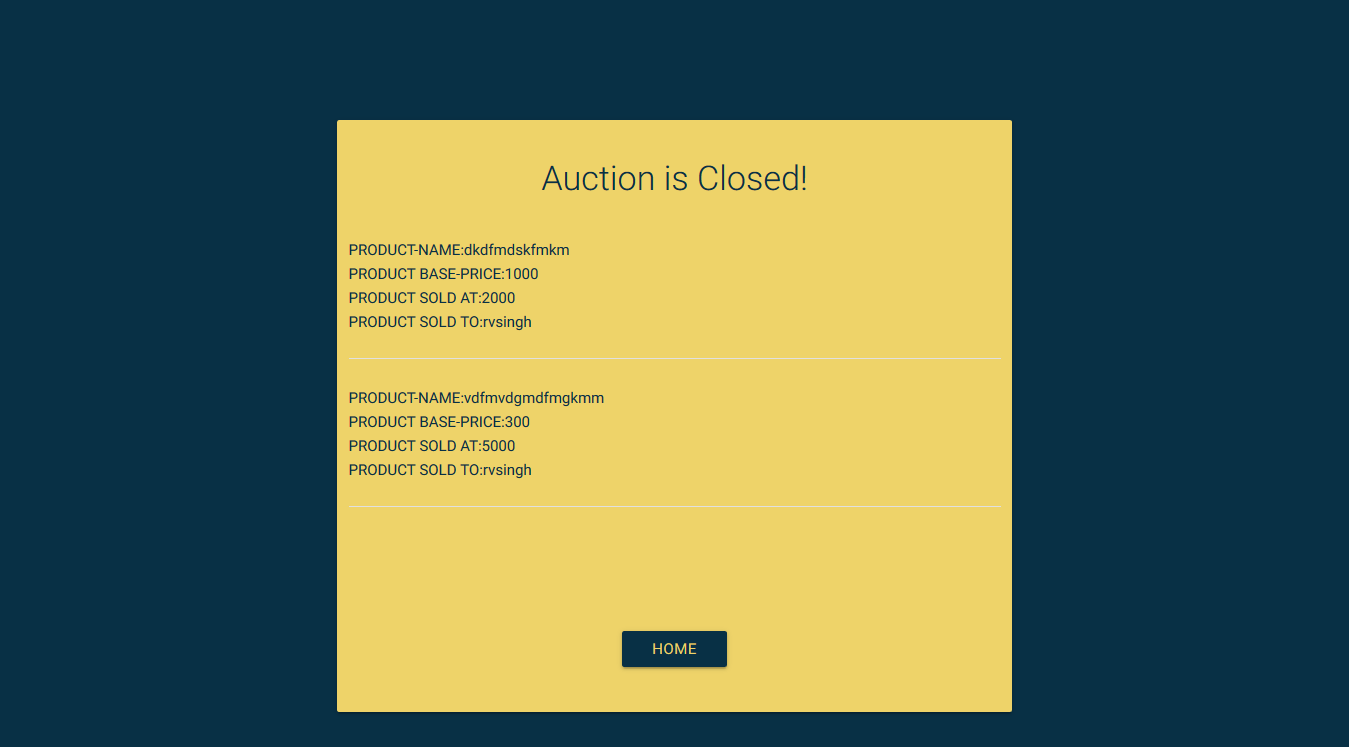
****

**Figure 6.6.1 T&C for Adding Product**

****

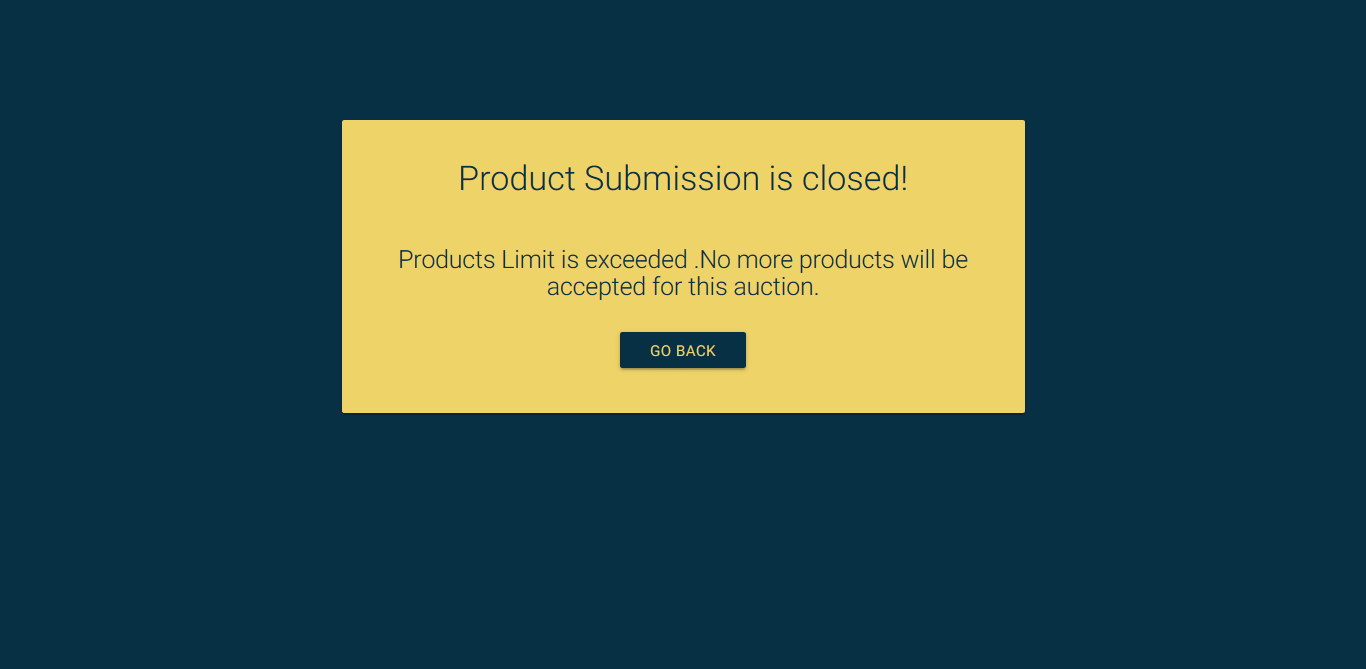
**Figure 6.6.2 Add Products Page**

**6.7 Auction Closed**

****

**Figure 6.7 Auction Closed Page**

**6.8 Product Submission Closed**

****

**Figure 6.8 Product Submission Page**

**CHAPTER : 7**

**LIMITATIONS**

**AND**

**FUTURE ENHANCEMENTS**

1. **Limitations and Future Enhancements**

* **Limitations**

1. Administrator is required to be present during the auction as the auction is not fully self-reliant .
2. Security issues as only Javascript Frameworks are used .

* **Future Enhancements**

1. More reliable auction with advanced event driven features.

2. Live Feed to admin about the ongoing auction.

**CHAPTER : 8**

**CONCLUSION**

1. **Conclusion**

Hereby, we declare that the functionality implemented in this system was performed by understanding all the modules.

This project aims to provide live auction mechanism which was fulfilled to a greater extent.All the UML designs where prepared before implementation and the coding was done accordingly. After the coding was completed, comprehensive testing was performed and the results were provided in the report. Unit Testing of all modules were done and later ,Integration Testing was also performed.

**CHAPTER : 9**

**BIBLIOGRAPHY**

1. **Bibliography**

**Web Tutorials:**

* nodeschool.io
* nodeguide.org
* nodetuts.com
* lynda.com
* Tutorialspoint.com for AngularJS , Node.JS and MongoDB
* mongoose.org

**Books:**

* ng-book: The Complete Book On Angular JS
* Angular JS: Up and Running - O’Riley Media
* The Def Guide to MongoDB
* MongoDB in action : Kyle Banker