Auction on Assets

OBJECTIVE:

The project is mainly aimed at automate the bidding process. In Auction automation the administrator is going to control the system, here he can add the categories of products or assets. Initially administrator will give minimum bid cost of the asset and other details like its category, name, bid date etc. Now the member of the system can see the initial bid details of that product, so that he can give his bid details just by clicking in its name.

Finally the administrator will check all the bids by login into the system. The member who has given the highest value will become the winner of the bidding process.

Purpose of the Project

The proposed system should serve the following feature:

- Automate the communication between the bidder and the admin.
- Adding the categories. Ex: Real Assets, Systems, Arts etc.
- Inserting bid under particular category.
- Provide updated details of the bids.
- Obtain the winner of bidding process.

Future Improvement

- Online auctioning can be included in the future through internet banking.
- User interaction will be easy in purchasing and selling of products.
- Enhancing Security by using thumb prints and voice recognition in future.
- Auctioning will be easier to every user through an SMS on the mobile.

Conclusion of the System

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for us as it provided practical knowledge of not only programming in ASP.NET and C#.net

web based application and no some extent Windows Application and SQL Server, but also about all handling procedure related with "Auction on Assets". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BACKGROUND:

Auctions are among the oldest economic institutions in place. They have been used since antiquity to sell a wide variety of goods, and their basic form has remained unchanged. In this dissertation, we explore the efficiency of common auctions when values are interdependent- the value to a particular bidder may depend on information available only to others-and asymmetric. In this setting, it is well known that sealed-bid auctions do not achieve efficient allocations in general since they do not allow the information held by different bidders to be shared.

Typically, in an auction, say of the kind used to sell art, the auctioneer sets a relatively low initial price. This price is then increased until only one bidder is willing to buy the object, and the exact manner in which this is done varies. In my model a bidder who drops out at some price can "reenter" at a higher price.

With the invention of E-commerce technologies over the Internet the opportunity to bid from the comfort of one's own home has seen a change like never seen before. Within the span of a few short years, what may have began as an experimental idea has grown to an immensely popular hobby, and in some cases, a means of livelihood, the online auction gathers tremendous response every day, all day. With the point and click of the mouse, one may bid on an item they may need or just want, and in moments they find that either they are the top bidder or someone else wants it more, and you're outbid! The excitement of an auction all from the comfort of home is a completely different experience.

The levels of comfort may rise in the near future but the rules to be followed remain the same. In fact may rise with the new technologies.

Society cannot seem to escape the criminal element in the physical world, and so it is the same with online auctions. This is one area where in a question can be raised as to how safe online auctions.

Problem in Existing System

- In existing system, the bidder sends his bid details manually not through online.
- New bid information cannot reach the bidder early. It should reach only via by news papers.

Solution of these Problems

So to avoid the problems in existing system, we have developed Auction on Assets application by introducing some more features.

- Providing information about the bid though online to the member.
- Giving permission to the member to bid on a particular product.
- Evaluating the member bids by the admin.

FUNCTIONAL REQUIREMENT:

Admin

- Administrator enters into the system by providing his login id and password
- Adding a new category to the system
- Requesting bids on asset of a category
- Evaluating all the posted bids given by the registered members
- Removing permission to a member
- Viewing all the posted bid information
- Viewing registered member information
- Change of password

Member

- Member enters into the system by providing his login id and password
- Member can view all the bids requested by the admin
- He can post a bid with his own bid value which is greater than actual bid cost.
- Modifying self data.
- Change of password

Proposed application should possess the following features

- Automate the communication between the bidder and the admin.
- Adding the categories. Ex: Real Assets, Systems, Arts etc.

- Inserting bid under particular category.
- Provide updated details of the bids.
- Searching bids in different categories.
- Posting the self bid value by the member.
- Obtain the winner of bidding process by the admin

NON-FUNCTIONAL REQUIREMENT:

Scalability and Availability: ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

Simplicity: ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

Manageability: ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

Power and Flexibility: Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.