|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **WEB APPLICATION FOR VEHICLE RESERVATION SYSTEM** | | | | |  |
|  | **A PROJECT REPORT** | | | | |  |
|  |  | **Submitted by** | | | |  |
|  | **DHAYANAND V**  **15CSR051** | | | | |  |
|  | ***in partial fulfilment of the requirements*** | | | | |  |
|  | ***for the award of the degree*** | | | | |  |
|  |  | | | ***of*** | |  |
|  |  | | | | |  |
|  | **BACHELOR OF ENGINEERING** | | | | |  |
|  |  | | | **IN** | |  |
| **COMPUTER SCIENCE AND ENGINEERING** | | | | | | | | | |
| **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** | | | | | | | |  | |
| **SCHOOL OF COMMUNICATION AND COMPUTER SCIENCES** | | | | | | | | |  |
|  |  | | finallogo.png | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  | **KONGU ENGINEERING COLLEGE** | | | | |  |
|  | (**Autonomous**) | | | | |  |
|  | **PERUNDURAI ERODE – 638 060** | | | | |  |
|  | **APRIL 2019** | | | | |  |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI ERODE-638060**

**APRIL 2019**

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled **WEB APPLICATION FOR VEHICLE RESERVATION SYSTEM** is the bonafide record of project work done by **DHAYANAND V (15CSR051)** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Computer science and Engineering of Anna University, Chennai during the year 2018-2019.

**SUPERVISOR HEAD OF THE DEPARTMENT**

(Signature with seal)

Date:

Submitted for the end semester viva voce examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI ERODE – 638060**

**NOVEMBER 2018**

**DECLARATION**

We affirm that the Project Report titled **“WEB APPLICATION FOR** **VEHICLE RESERVATION SYSTEM”** being submitted in partial fulfilment of the requirements for the award of Bachelor of Technology is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**Date:**

**DHAYANAND V**

**15CSR051**

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Date: Name and Signature of the Supervisor with seal

**ABSRTACT**

An all-inclusive vehicle rental management tool is necessary for seamless operations of an enterprise rental company. Centralized inventory management is required to alert expiry of vehicle insurance, adherence towards scheduled maintenance and notification towards ageing vehicles. This also helps to oversee vehicle in use at different branches of the company and be supplied with revised models of the vehicles from the central inventory to the branch. This also helps to avert non-availability of vehicles at a branch store at any given time. The solution developed will address the objective in a holistic manner and will have all the features and functionalities which shall let the fleet management company be able to record and update the vehicle details along with their insurance details. The application will also run batch jobs to alert the admin on the expiry of the vehicle insurance to ensure the Vehicles moving out are 100% checked and approved.

**ACKNOWLEDGEMENT**

First and foremost we acknowledge the abundant grace and presence of almighty throughout different phases of the project and its successful completion.

We wish to express our extreme gratefulness to our beloved Correspondent **Thiru.A.VENKATACHALAM, B.Sc.,** and all the trust members of Kongu Vellalar Institute of Technology Trust for providing all the necessary facilities to complete the project successfully.

We express our deep sense of gratitude to our beloved Principal **Prof.S.KUPPUSWAMI B.E., M.Sc (Engg)., Dr.Ing (France).,** for providing us an opportunity to complete the project.

We express our gratitude to **Dr.R.R.RAJALAXMI M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering, for her valuable suggestions.

We are thankful to our project coordinator **Mrs.N.SASIPRIYAA M.E.,** Assistant Professor for her valuable guidance and support to complete our project successfully.

We are highly indebted to **Mrs.N.SASIPRIYAA M.E.,** Assistant ProfessorDepartment of Computer Science and Engineering for her valuable supervision and advice for the fruitful completion of the project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER No.** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | iv |
|  | **ACKNOWLEDGEMENT** | v |
|  | **LIST OF FIGURES** | viii |
|  | **LIST OF ABBREVATIONS** | ix |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1 SYSTEM STUDY | 2 |
|  | 1.2 OBJECTIVE | 3 |
|  | 1.3 SCOPE | 4 |
| **2** | **GENERAL DESCRIPTION** | 5 |
|  | 2.1 PROJECT PERSPECTIVE | 5 |
|  | 2.2 USER CHARACTERISTICS | 6 |
|  | 2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS | **7** |
| **3** | **SYSTEM REQUIREMENTS** | 8 |
|  | 3.1 FUNCTIONAL REQUIREMENTS | 8 |
|  | 3.2 NON-FUNCTIONAL REQUIREMENTS | 9 |
|  | 3.3 USER INTERFACES | 10 |
| **4** | **DETAILED DESIGN** | 12 |
|  | 4.1 ARCHITECTURE DESIGN | 12 |
|  | 4.2 USECASE DIAGRAM | 13 |
|  | 4.3 ACTIVITY DIAGRAM | 13 |
|  | 4.4 DATABASE DESIGN | 15 |
|  | 4.5 OUTPUT DESIGN | 15 |
| **5** | **RESULTS AND DISCUSSION** | 17 |
| **6** | **TESTING** | 18 |
| **7** | **CONCLUSION AND FUTURE WORK** | 20 |
| **8** | **APPENDICES** | 21 |
| **9** | **REFERENCES** | 31 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 4.1.1 | Architecture diagram of the application | 12 |
| 4.2.1 | Usecase diagram | 13 |
| 4.3.1 | Activity diagram | 14 |
| 4.3.1 | Entity relationship diagram | 15 |
| 4.3.2 | Personal details table structure | 16 |
| 4.3.3 | Academic details table structure | 16 |

**LIST OF ABBREVATIONS**

**AJAX** Asynchronous JavaScript And XML

|  |  |
| --- | --- |
| **CSS** | Cascading Style Sheets |
| **JDBC** | Java Database Connectivity |
| **HTML** | Hyper Text Markup Language |
| **HTTP** | Hypertext Transfer Protocol |
| **JS** | JavaScript |
| **HP** | HypertextPreprocessor |
| **SQL** | Structure Query Language |

**CHAPTER 1**

1. **INTRODUCTION**

**1.1 SYSTEM STUDY**

The purpose of this project is to build an “vehicle reservation system”, a place for enterprise rental company to maintain the vehicle details and to alert expiry of vehicle insurance, adherence towards scheduled maintenance & notification towards ageing vehicles. The system consists in a web-portal where registered users on logging in they can able to register new vehicle details, view registered vehicle details, update vehicle details and search vehicle details. Registration of users is preceded by a “admin registration”, after successful registration they can able to login to the system. Without this confirmation, a user cannot access the functionality of the portal. Auctions have a name, a description, users cannot place bids when the auction interval (start - end period) ends, but. Moreover, administrators have the possibility to view the registered users , items and to view the bidding status for all the product based on the category given by the administrator. The system is realized with a 3-tier architecture. It consists of a relational database that store the information regarding items, users, auctions and categories of auction; an application server that cares about the business logic of the system and the presentation layer that consists in the web browser where users can interact with the system. With such architecture, the database is never directly accessed: for example administrators can change the data stored in the database without connecting directly to it but using their own browser.

**1.2 OBJECTIVE**

The problem with public auction is that the participation of the general public is very limited. The aim of the project is to socialize the auction so that people from far & wide and even across the continent can participate in it. The "E-Auction" site is developed with a vision to wipe out the inherent problems of "Conventional Auction House". "E-Auction" is designed in such a way that it is as user friendly as possible. So any aspiring bidder or seller can visit the site and engage in bidding with least effort.

The salient features of the site are as follows:

1. Paperless Auction System
2. It's accessible to everyone, at any time no matter where they are
3. Reliable user validation & checking.
4. Easy online settlement.

**1.2 SCOPE**

"E-Auction" is an online auction web site aimed at taking the auction to the fingertips of aspiring bidders there by opening up the doors of the "OPEN Auction House' to a wider cross section of Art Lovers and Antique Collectors. This site also acts as an open forum where buyers and sellers can come together and exchange their products. The site makes sure that the sellers get a fair deal and buyers get a genuine product. This project contains the user and admin module, the user can be either seller or buyer. The seller can put their product for sales and view their product’s bidding status occasionally till their bidding end date. When the product’s bid date ends, the product is sold to the particular user whom bidding price is very high. The buyer can search for a product to buy based the category they click all the items in that category will be displayed with the last bid price. When the buyer clicks on the desired product it moves on to the bidding page where all the details like remaining number of days to end the bidding for that product, short description about the product will be displayed and the buyer can bid for the product it will updated as the last bid price of that product. The admin can login into the application and view the bidding status of all the products and user details can also be viewed.

Online application of the whole system helps easy access anywhere. The time taken for process completion is now largely reduced. After bidding the database is automatically updated to the last buyer’s bid price. The database is managed by MySQL, so data will be easily accessed and retrieved. Data cannot be viewed or edited by unauthorized personnel. So, this automated and computerized system is safe, fast and user friendly. The front end designed using java server page that is connected to the database using servlet.

**CHAPTER 2**

**2. GENERAL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

The proposed computerized "E-Auction" site has made auction process simple. The only pre-condition is that the user must register and authenticate before he/she can take part in the bidding process. The system uses HTTP forms authentication which creates a session cookie for any signed in user. Throughout the span of the session the cookie remains valid until the user logs out.

An auction house needs to have products to auction, so in the proposed system this is done using product registration module. The module is open to user who is registered sellers and they need to authenticate before they register any product. The user can put a product for sales or they can buy a product through bidding.

Admin if this page can login into the page and view the bidding status of all the products and they can also view all the user details.

**2.2 USER CHARACTERISTICS**

The product should be user friendly; hence the user should not need any software and hardware knowledge. They are assumed to have basic knowledge of computers. Administrators of the system should have more knowledge of internal modules of the system and are able to rectify small problems that may arise due to disk crashes, power failures and other catastrophes. Friendly user interface, user guide must be sufficient to educate the users on how to use this product without any problems or difficulties.

**2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS**

* The information of users who have registered are stored in database.
* Apache Tomcat 7.0 Web Server.
* It can be operated at any time.
* User can bid for the product under auction before the end date.

**CHAPTER 3**

**3. REQUIREMENTS**

**3.1 FUNCTIONAL REQUIREMENTS**

**3.1.1 User Module**

**Prerequisite (**user signed in) for all requirements below

**Requirement ID** R1.01.01

**Title** Login in module

**Description** This action is done to login to the site

**Requirement ID** R1.01.02

**Title** New User Register

**Description** This action is done to enter the details of new user.

**Requirement ID** R1.01.03

**Title** Main Module (Put for Sale)

**Description** This action is done to put products for sales in auction

**Requirement ID** R1.01.04

**Title** Main Module (Search Product)

**Description** This action is done to search products for bidding

**Requirement ID** R1.01.05

**Title** Main Module (Bidding Center)

**Description** This action is done to bid the products at highest price.

**Requirement ID** R1.01.05

**Title** Main Module (My Purchase)

**Description** This action is done to display the products bought in bidding

**Requirement ID** R1.02.03

**Title** Main Module (My Profile)

**Description** This action is done to view and edit user profile

**3.1.2 Admin Module**

**Prerequisite (**user signed in) for all requirements below

**Requirement ID** R1.01.01

**Title** Login in module

**Description** This action is done to login to the site

**Requirement ID** R1.01.05

**Title** Main Module (View Users)

**Description** This action is done to view personal details of all users.

**Requirement ID** R1.01.05

**Title** Main Module (View Bidding Status)

**Description** This action is done to view bidding status of all products.

**3.2 NON-FUNCTIONAL REQUIREMENTS**

**3.2.1 Error Handling**

* E-Auction product shall handle expected and non-expected errors in ways that prevent loss in information.

**3.2.2 Performance Requirements**

* The system accommodates high number of products and their prices.
* Responses to view information shall take no longer than 5 seconds to appear on the screen.

**3.2.3 Safety Requirements**

* Maintainability
* Reliability.

**3.2.4 Security Requirements**

* System will use secured database.
* Product prices are updated only with user’s knowledge.

**3.3 USER INTERFACE**

**3.3.1 Hardware Requirements**

Hard disk: 250 GB and above

RAM: 2 GB and above

Processor: i3 and above

**3.3.2 Software Requirements**

Operating system: windows 7 and above

Web server: APACHE TOMCAT-8

RDBMS: MySQL

**3.3.3 Technologies Used**

**JSP**

Java Server Pages (JSP) is a Java standard technology that enables you to write dynamic, data-driven pages for your Java web applications. JSP is built on top of the Java Servlet specification. The two technologies typically work together, especially in older Java web applications. From a coding perspective, the most obvious difference between them is that with servlets you write Java code and then embed client-side markup (like HTML) into that code, whereas with JSP you start with the client-side script or markup, then embed JSP tags to connect your page to the Java backend.

JSP is also closely related to JSF (JavaServer Faces), a Java specification for building MVC (model-view-controller) web applications. JSP is a relatively simpler and older technology than JSF, which is the standard for Java web frameworks like Eclipse Mojarra, MyFaces, and PrimeFaces. While it is not uncommon to see JSP used as the frontend for older JSF applications, Facelets is the preferred view technology for modern JSF implementations.

While JSP may not be your first choice for building dynamic web pages, it is a core Java web technology. JSP pages are relatively quick and easy to build, and they interact seamlessly with Java servlets in a servlet container like Tomcat. You will encounter JSP in older Java web applications, and from time to time you may find it useful for building simple, dynamic Java web pages. As a Java developer, you should at least be familiar with JSP.

This article will be a quick introduction to JavaServer Pages, including the JSP Standard Tag Library (JSTL). Examples show you how to write a simple HTML page, embed JSP tags to connect to a Java servlet, and run the page in a servlet container.

**SERVLET**

A **servlet** is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.

The javax.servlet and javax.servlet.http packages provide interfaces and classes for writing servlets. All servlets must implement the Servlet interface, which defines life-cycle methods. When implementing a generic service, you can use or extend the Generic Servlet class provided with the Java Servlet API. The HttpServlet class provides methods, such as doGet and doPost, for handling HTTP-specific services.

**BOOTSTRAP**

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a front-end framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins. It facilitates you to create responsive designs. It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap. It facilitates users to develop a responsive website. It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

**JQUERY**

jQuery is a fast, small, cross-platform and feature-rich JavaScript library. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers.

The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.

jQuery is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as write less do more? because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. jQuery is a small, fast and lightweight JavaScript library. jQuery is platform-independent. jQuery means "write less do more". jQuery simplifies AJAX call and DOM manipulation.

**ECLIPSE**

Eclipse and MyEclipse for the programming part of the project we used Eclipse, that is a very powerful open source integrated development environment (IDE). This IDE offers several services to the developer: it has compiler aware editing; syntax errors are highlighted when they are made, as are simple semantic errors such as missing declarations. Eclipse supports method completion, shows class interfaces concisely in graphical notation and supports interactive exploration of a program, through features such as fly-over name resolution. In addition, Eclipse supports Software Engineering principles such as packaging, debugging, testing, refactoring and versioning [2]. Another very important advantage of using Eclipse is that several plug-ins can be installed on it, like MyEclipse, that is particularly useful when working with Enterprise JavaBeans.

**Database and JDBC**

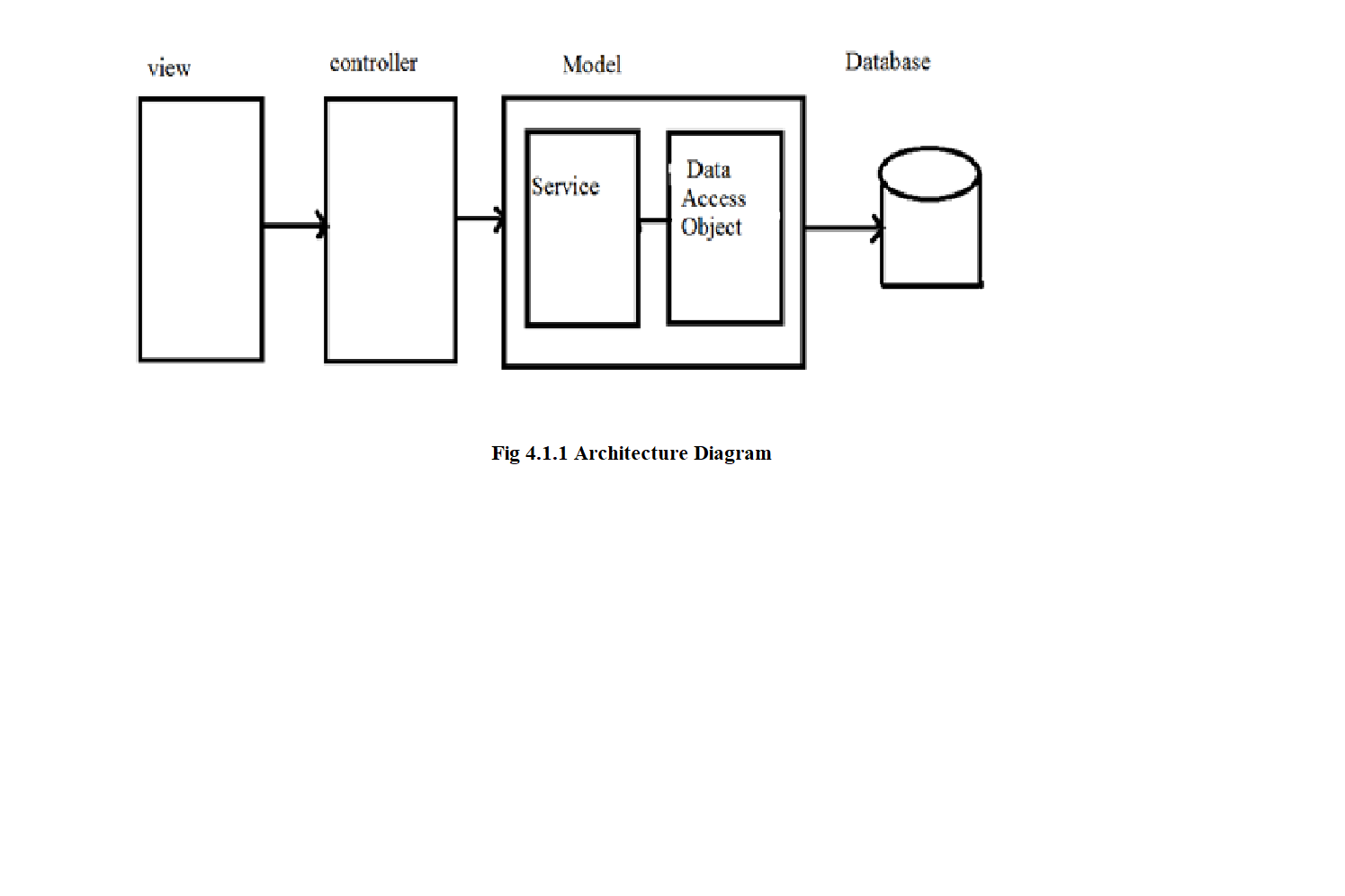
As database for the project we used PostgreSQL, that is an open-source relational database. A relational database is a type of database management system (DBMS) that stores data in the form of related tables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. The most common options when choosing a DBMS are Oracle [9], MySQL [7] and PostgreSQL [10].

**CHAPTER 4**

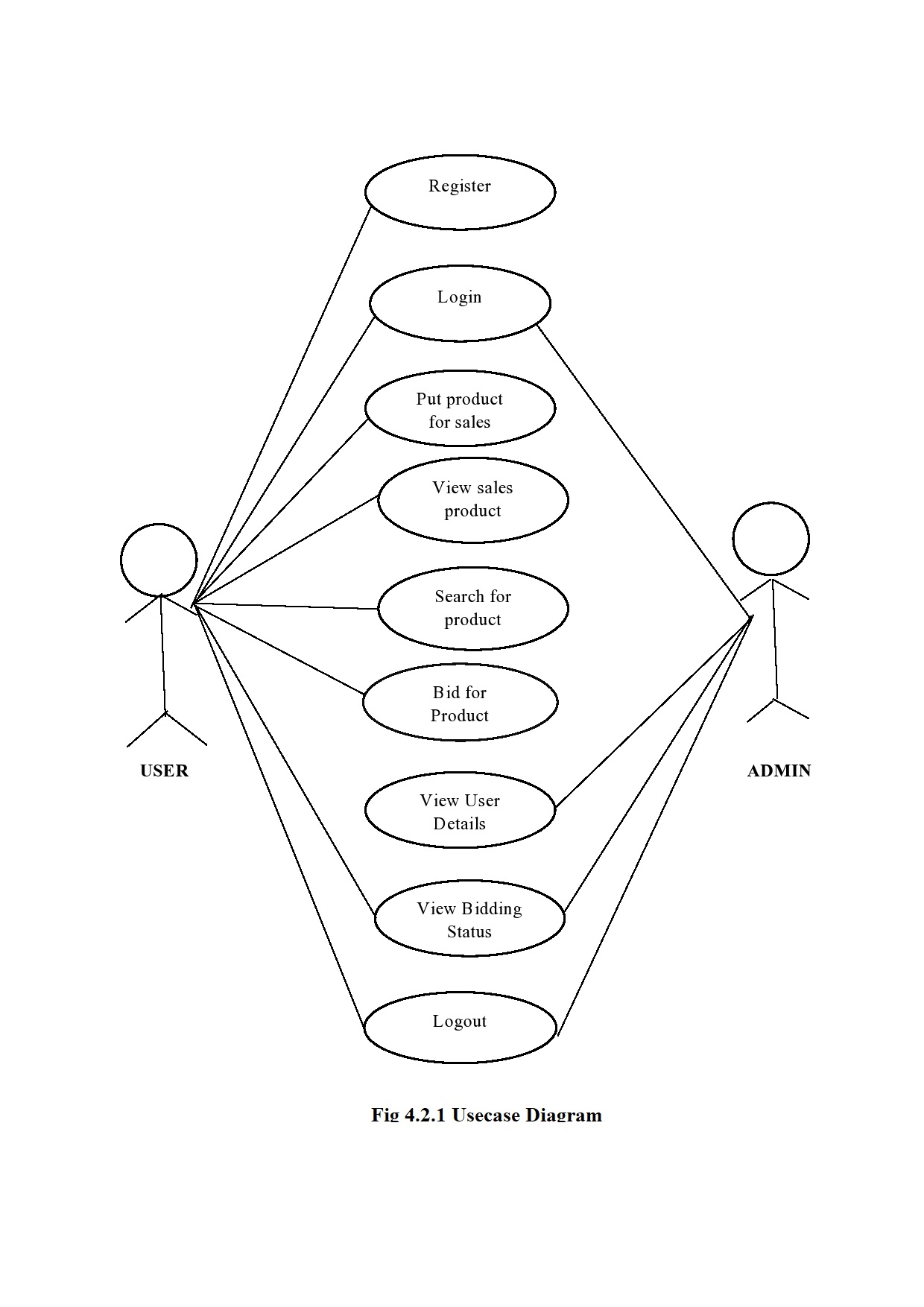
**4. DETAILED DESIGN**

**4.1 ARCHITECTURAL DESIGN**

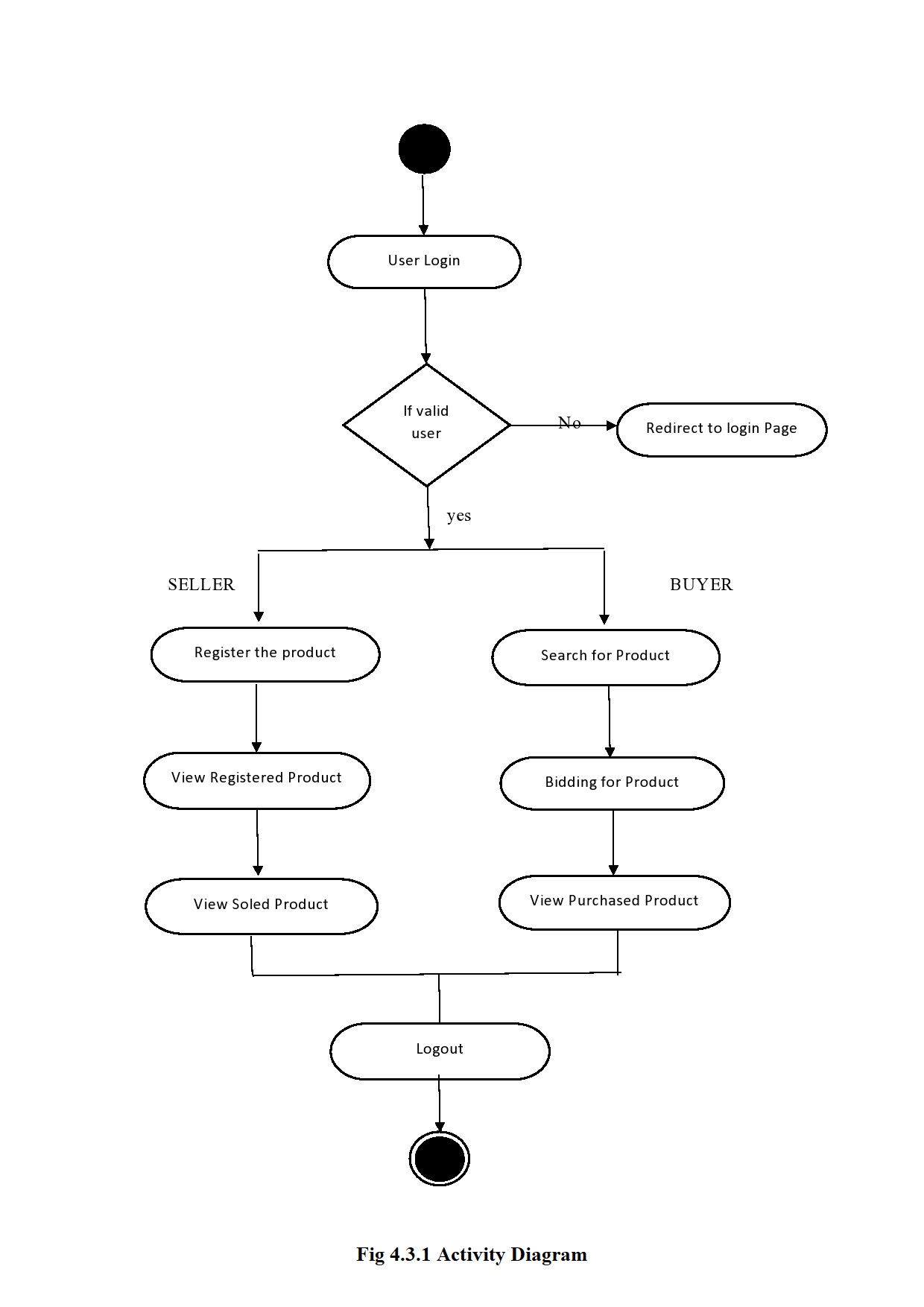
The architecture for any application is broken into three separate logical layers, each with a well - defined set of interfaces. The first tier is referred to as the presentation layer and typically consists of graphical user interface of some kind. The middle tier, or business layer, consists of application or business layer and the third layer- the data layer contains the data that is needed for the application. The middle tier is basically the code that the user calls upon to retrieve the desired data. The presentation layer then receives the data and formats it for display. This separation of application logic from the user interface adds enormous flexibility to the design of application. The third tier contains the data that is needed for the application.

****

**4.2. USECASE DIAGRAM**

****

**4.3. ACTIVITY DIAGRAM**

****

**4.4 DATABASE DESIGN**

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. DBMS allow data to be protected and organized separately from other resources. Database is an integrated collection of data. The most significant form of data as seen by the programmers is data as stored on the direct access storage devices. This is the difference between logical and physical data.

Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information.

The organization of data in database aims to achieve three major objectives: -

* Data integration.
* Data integrity.
* Data independence

The proposed system stores the information relevant for processing in the MS SQL SERVER 2000 database. This database contains tables, where each table corresponds to one particular type of information. Each piece of information in table is called a field or column. A table also contains records, which is a set of fields. All records in a table have the same set of fields with different information. There are primary key fields that uniquely identify a record in a table. There are also fields that contain primary key from another table called foreign keys.

Database design is the organization of data according to database model. The database

design consists of database tables

1. User Details table
2. Put for Sales table
3. Bidding table
4. Feedback Table

**User Details**

User Details table consists of fields such as name, password, userid, phone number, address.

The details entered by the user is saved in database for any further updates or changes in their personal details can updated.

**Put for Sales table**

Put for Sales table consists of fields such as name, product name, product id, short description, detail description, starting bid price, bid end date.

The details entered by the seller is saved in database and it is updated when the bid price is given it is updated dynamically as the last bid price.

**Bidding table**

Bidding table consists of fields such as Bidder name, product id, bid price, bid date.

The details entered by the buyer is saved in database and it is updated when the bid price is given it is updated dynamically as the last bid price. Bidding details for all the products.

**Feedback Table**

Feedback table consists of fields such as Bidder User name, user id, phone number, feedback.

The details entered by the user is saved in database and it is viewed by the admin.

**4.5. OUTPUT DESIGN**

The output of the online auction system consists of the output in favour to the online sales and purchase of the products. The user personal details and the bidding status of all the products are viewed by the administrator

**CHAPTER 5**

**5. TESTING**

**5.1. UNIT TESTING**

A Unit corresponds to a screen /form in the package. Unit testing focuses on verification of the corresponding class or Screen. This testing includes testing of control paths, interfaces, local data structures, logical decisions, boundary conditions, and error handling. Unit testing may use Test Drivers, which are control programs to co-ordinate test case inputs and outputs, and Test stubs, which replace low-level modules. A stub is a dummy subprogram.

**5.2. REGRESSION TESTING**

Each modification in software impacts unmodified areas, which results serious injuries to that software. So the process of re-testing for rectification of errors due to modification is known as regression testing.

**Installation and Delivery:**

Installation and Delivery is the process of delivering the developed and tested software to the customer. Refer the support procedures

**Acceptance and Project Closure:**

Acceptance is the part of the project by which the customer accepts the product. This will be done as per the Project Closure, once the customer accepts the product; closure of the project is started. This includes metrics collection, PCD, etc.

**5.3 VALIDATION**

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. This application is validated using jquery validation.The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment.

By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions. Apart from these tests, there are some special tests conducted which are given below:

Peak Load Tests: This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new software for the agency is capable of handling the demands at the peak time.

Storage Testing: This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.

Performance Time Testing: This test determines the length of the time used by the system to process transaction data.

**CHAPTER 6**

**6. RESULTS AND DISCUSSIONS**

In business, there are situations were one get cheated by the people they trust. One way to safeguard their properties is to rely on technology. This project enables a person to have accurate results and to maintain each and every details securely. This work has been successfully completed and has been capable of maintaining trust between the users and the auction system and also, to restrict undesired bidding behaviours like shilling at runtime. Furthermore, it would allow all the parties in the bidding to have access to real-time reporting of events. The reports generated gives an outlook of a shop’s status and allows them to improve at tragic times. By automating by technology a lot of man power is reduced and allows them to concentrate on enhancing their economic growth of the company. Improving technology and implementing them can result in concise and prosperity in work environment.

This online system is used for customers to view all products that is up for auctioning and for the customers to upload their products for auctioning. The system has a platform for customers to post their bids and online advertising of products. The system gives bidders access to view auction details and detect shilling biddings. The system makes available the bidders history and maintain a robust database for managing the auction sales. It also maintains a database of bidders information.

**CHAPTER 7**

**7. CONCLUSION AND FUTURE WORK**

In this project, we addressed the problem of erroneous calculations in purchase and sales account of a retail store. One problem is the existing work was, human errors and accurate results cannot be computed due to lack of knowledge at some unintentional cases. Thus our work provided an optimal solution that can provide a platform to proceed calculations accurately which was integrated in a desktop based application. When this system proves to be flexible to the users and achieves to be user friendly then it can go through a series of advance steps that might be handy to use for all the users. The future work may comprise of paying bill through online transactions and paying taxes being automated. If more numbers are benefited, then the details can be stored in cloud and lead to peer-to-peer interactions. This system can be modified in a further for audit computation and online transactions can be added if it is to be implemented as web application. Many different adaptations, test and experiments have been left for the future due to lack of time. Future work concerns deeper analysis of particular mechanism, new proposals to try different methods or simply curiosity. For further better improvements it relies on user’s needs and technical advisor’s optimized level of thinking.

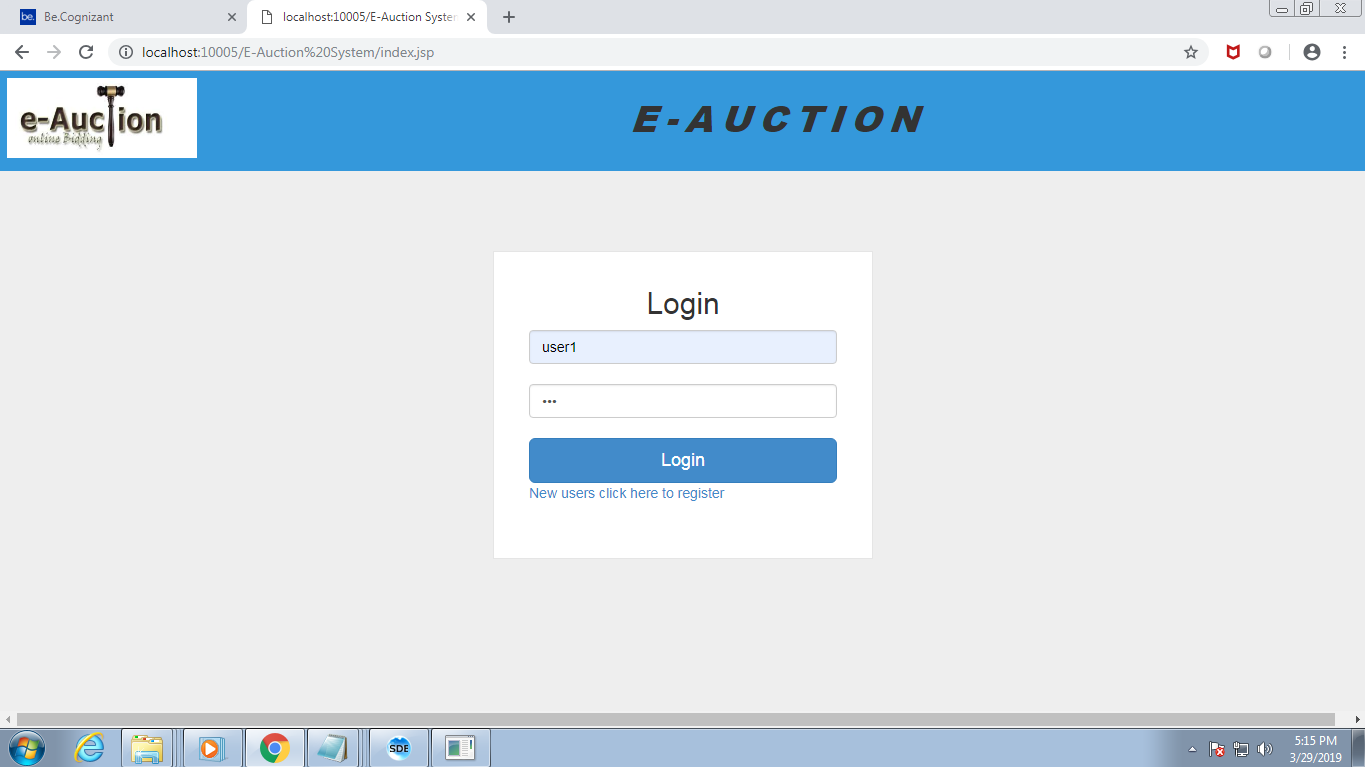
**CHAPTER 8**

**8. APPEDICES**

**8.2 SCREENSHOTS**

**8.2.1 LOGIN MODULE**

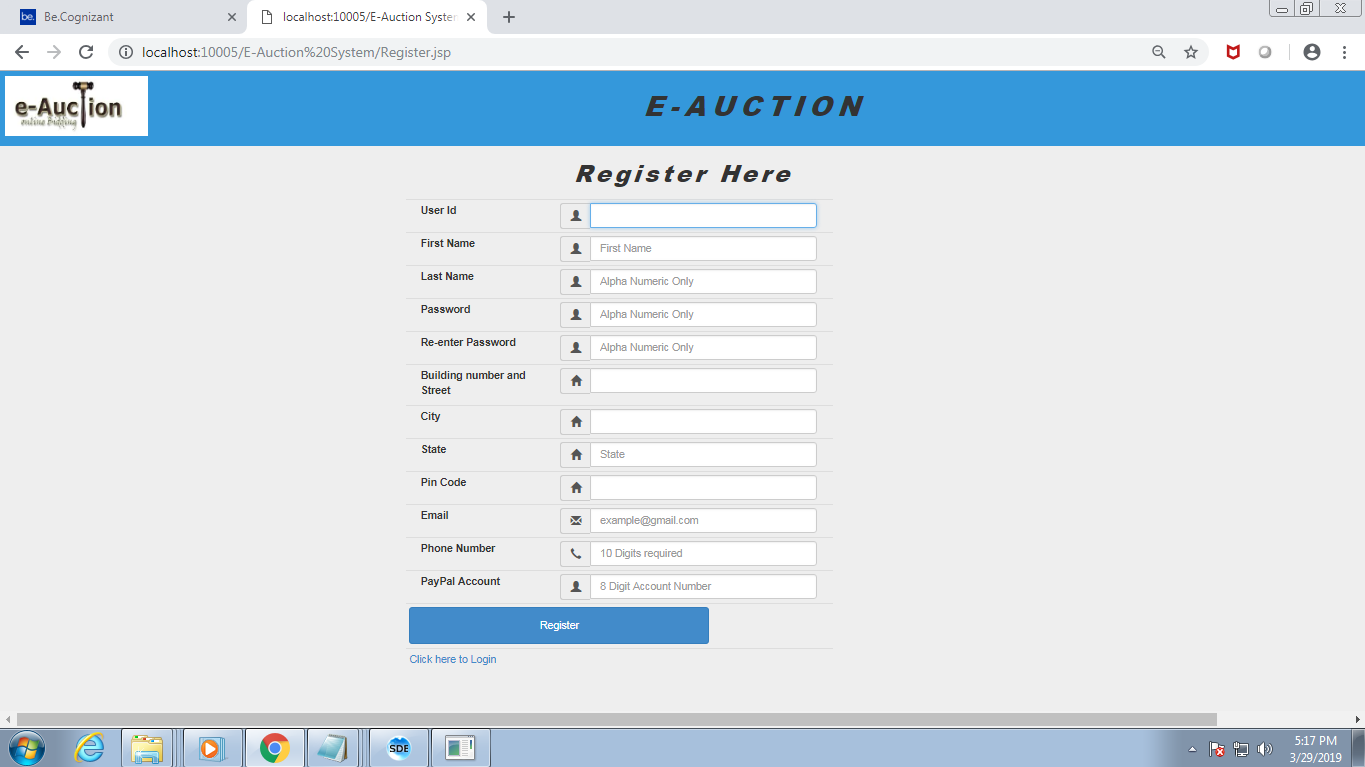
User can login with his user id and password and check all the functionality from the menu. Refer screen print below for the User login.

****

**Figure 8.2.1: Login Module**

**8.2.2 NEW USER REGISTRATION:**

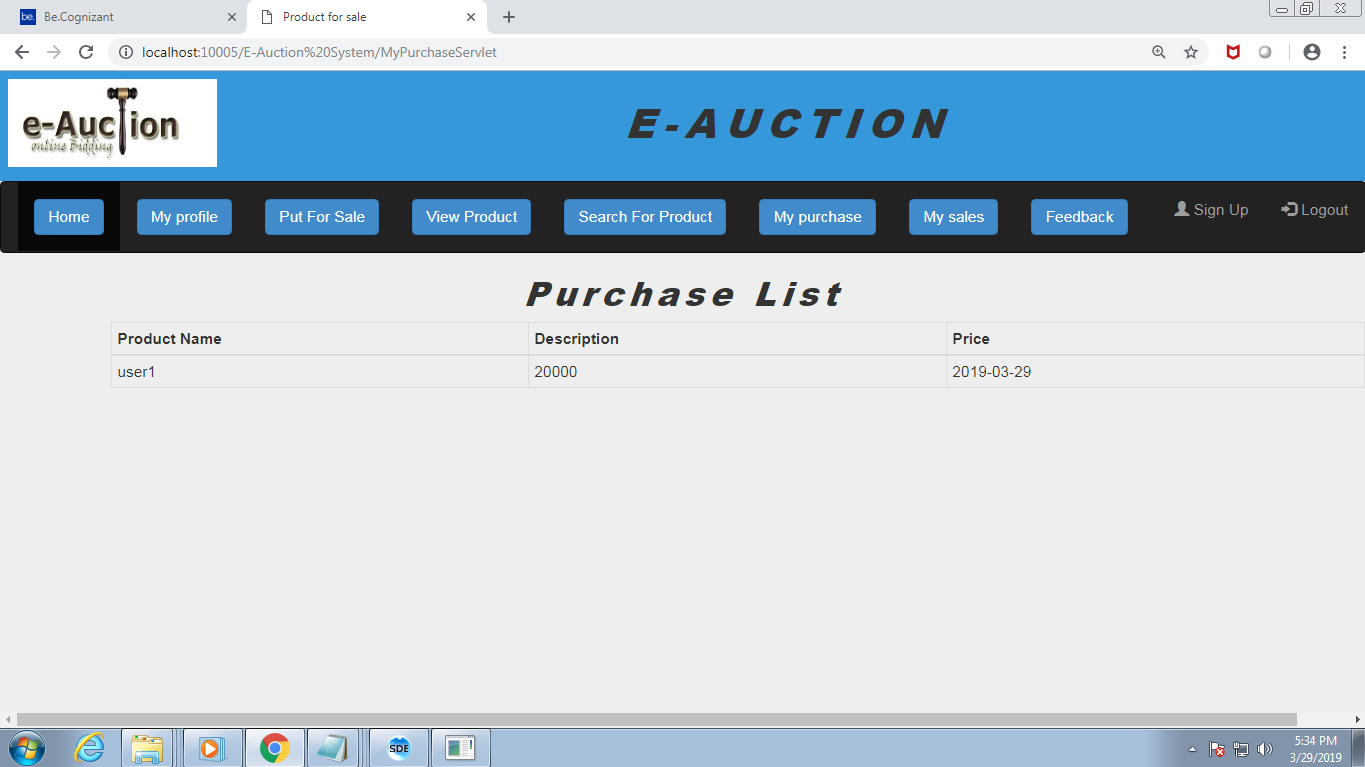
New user can be enrolled to the application by clicking **Register** link. User will be guided to fill registration form. User has to fill in all the mandatory field, not adhering to which lead to error message dialog to pop up. User can cancel the pop up by clicking **ok** in the dialog box and fill in the required field. When user leaves User name field blank, error dialog pops up. If the user enters all the correct credentials and click submit button user will redirected to welcome page.



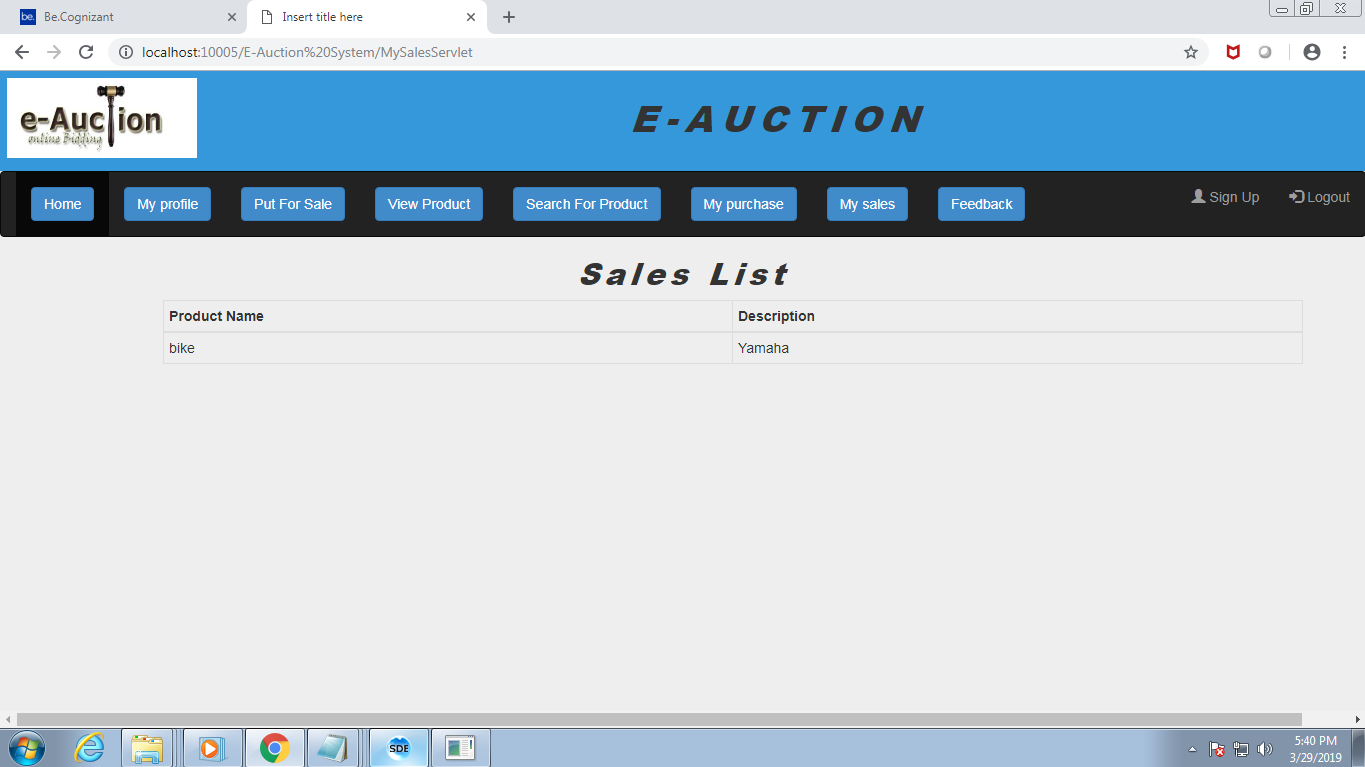
**Figure 8.2.2: New User Registration**

**8.2.3 MY PURCHASE AND MY SALE:**

This module helps the user in viewing their products that are sold to other bidders and the products that they have bought in auction.



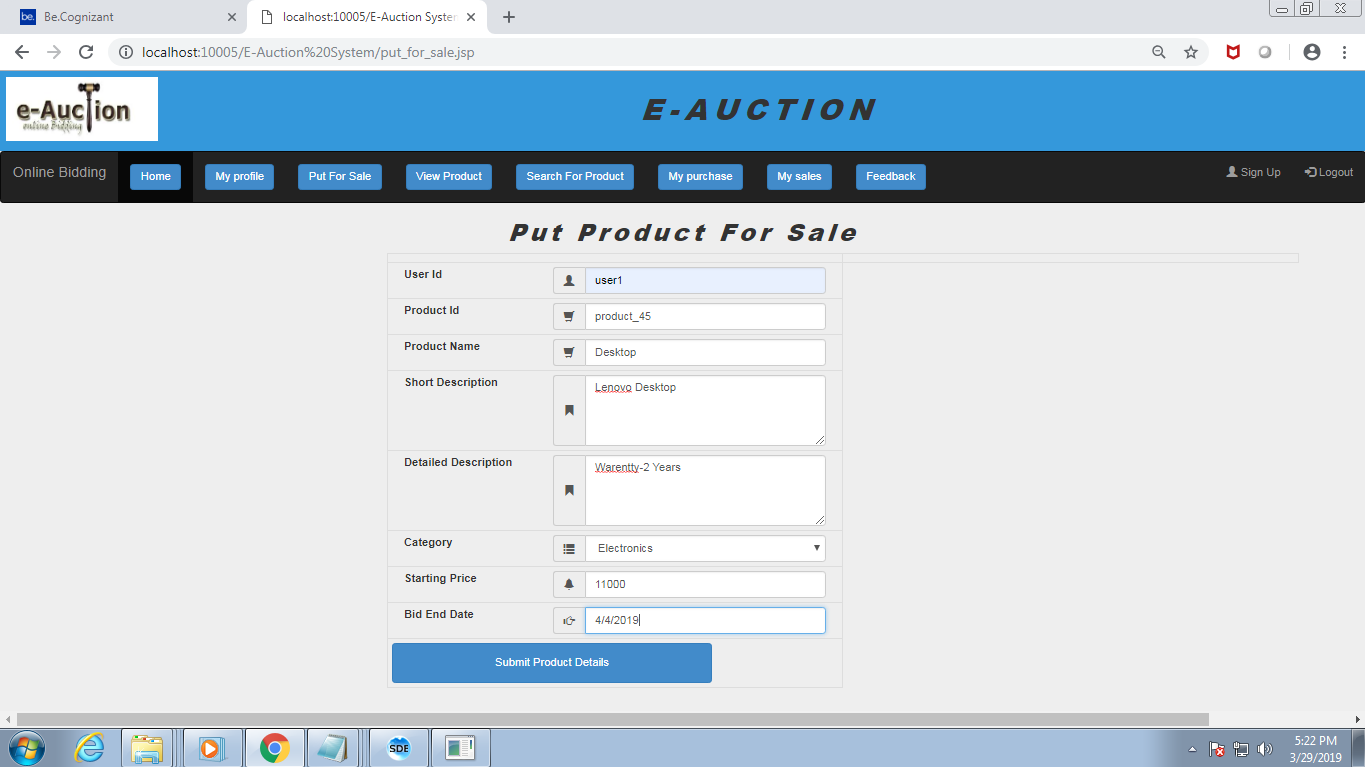
**Fig: 8.2.3 Purchase page**

****

**Fig: 8.2.3 Sales page**

**8.2.4 PUT FOR SALE:**

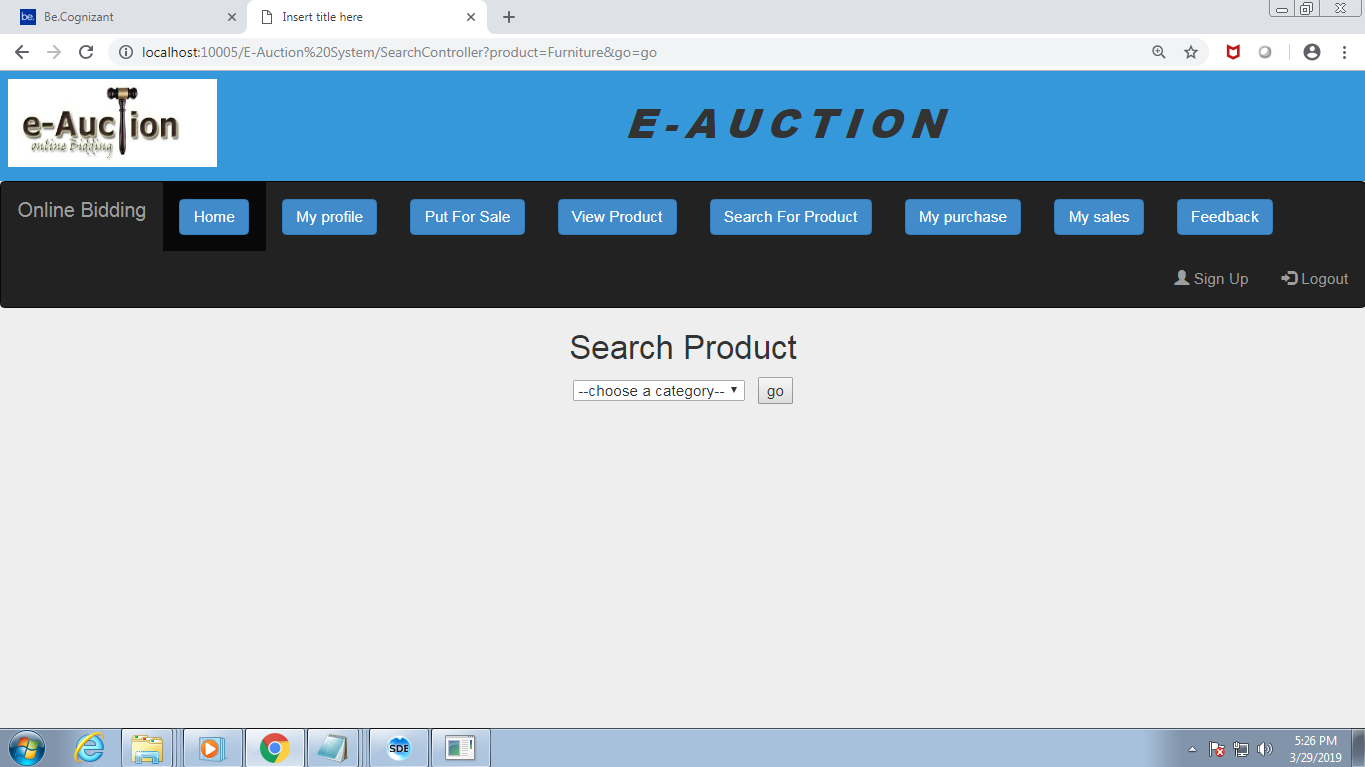
By clicking put for sale link user can put his item to be sold by maximum bid. After user clicks submit button after filling all the mandatory fields like category from drop box ,choose a starting price for his product , when he want to close the bid etc, he will be redirected to welcome page with confirmatory message.

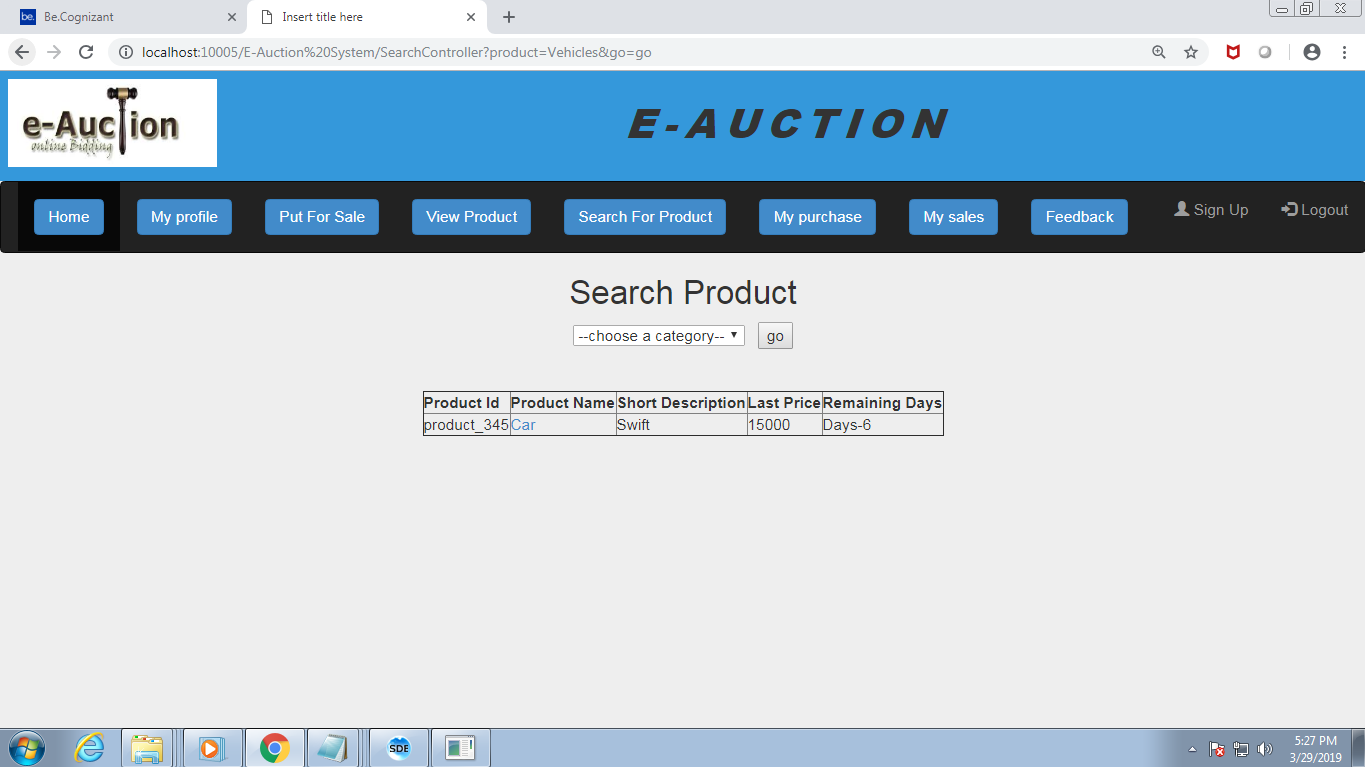
****

**Fig: 8.2.4 Put product for sale page**

**8.2.5 SEARCH PRODUCT:**

By clicking search product he can buy any product by searching the product in choose a category drop box. User will be provided with the list of all the products available in that category, he/she can click hyper link of any product he/she likes . User will be redirected to Bidding Center to bid for the product which should be greater than the starting price. His bid will be posted in the bidding table with bid closing date .

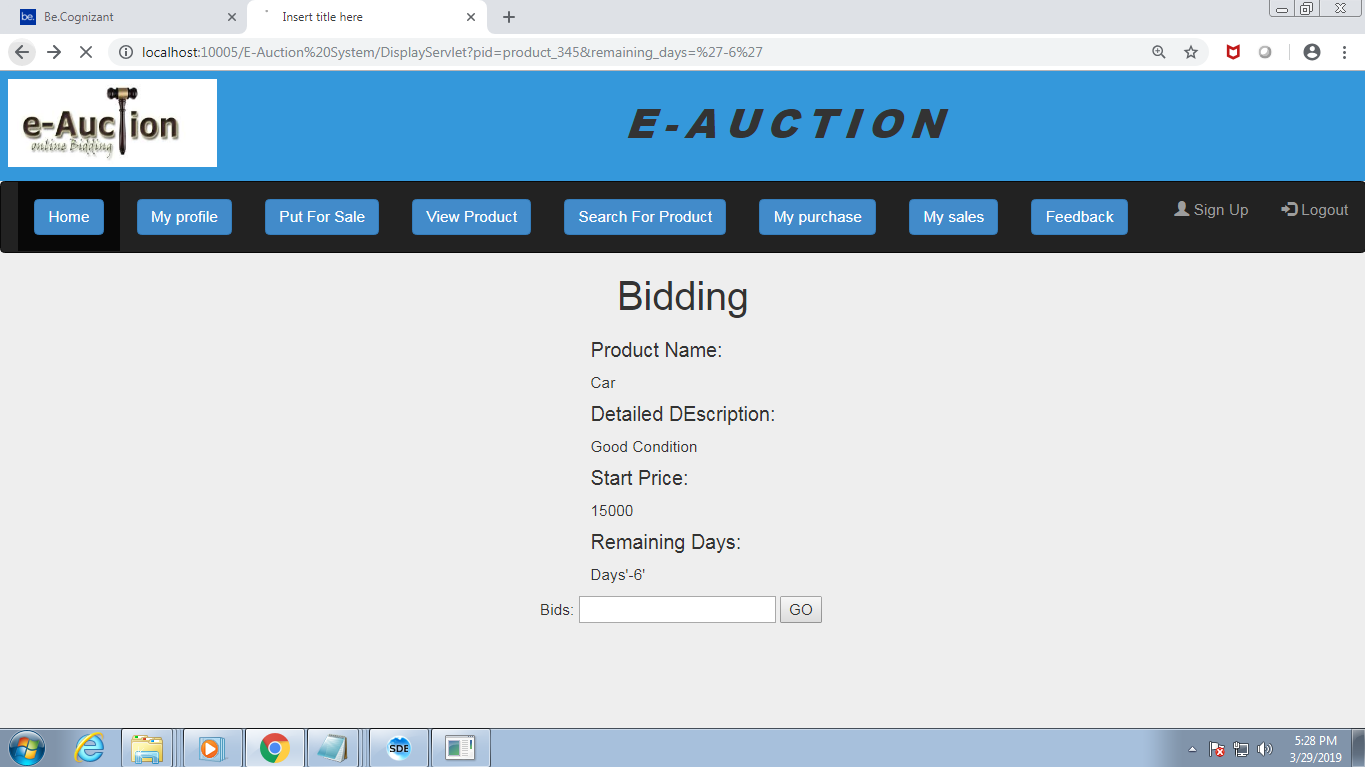
.

****

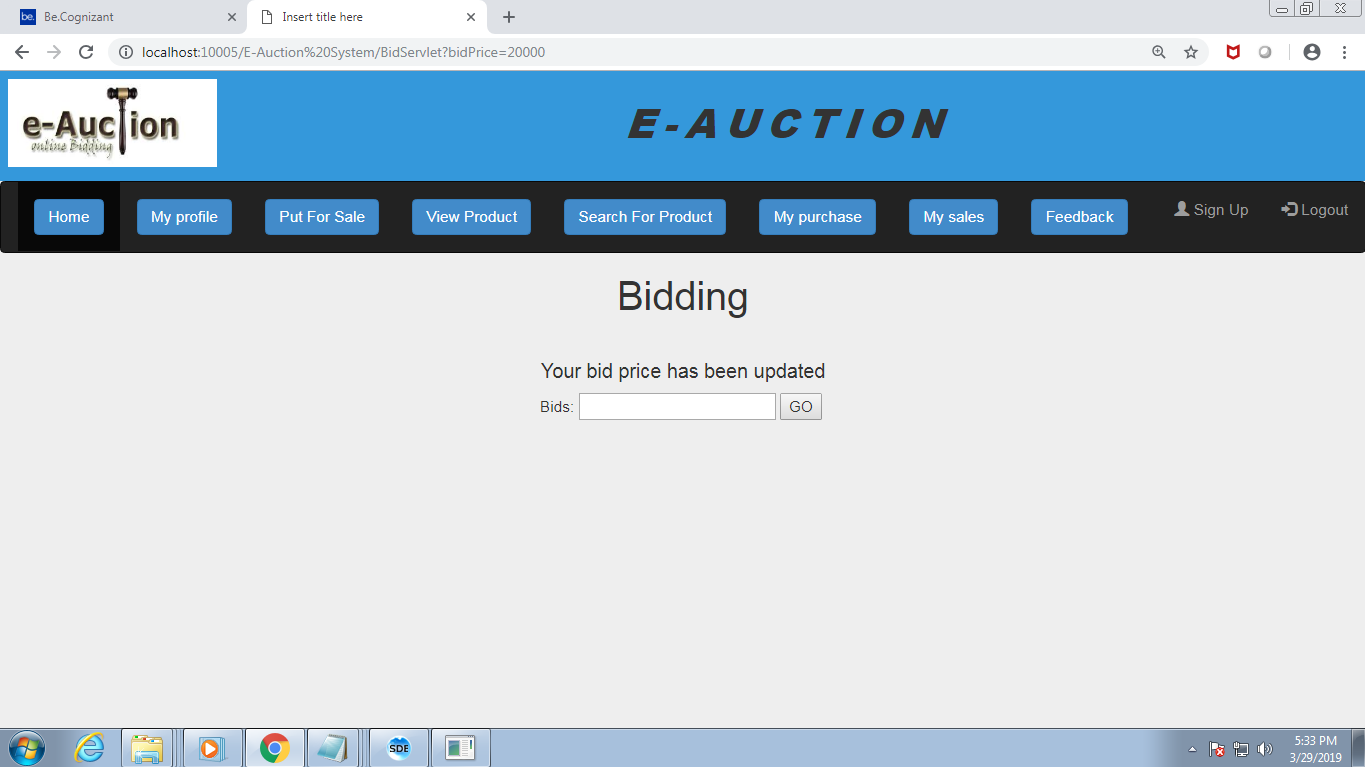
**Fig: 3.5**

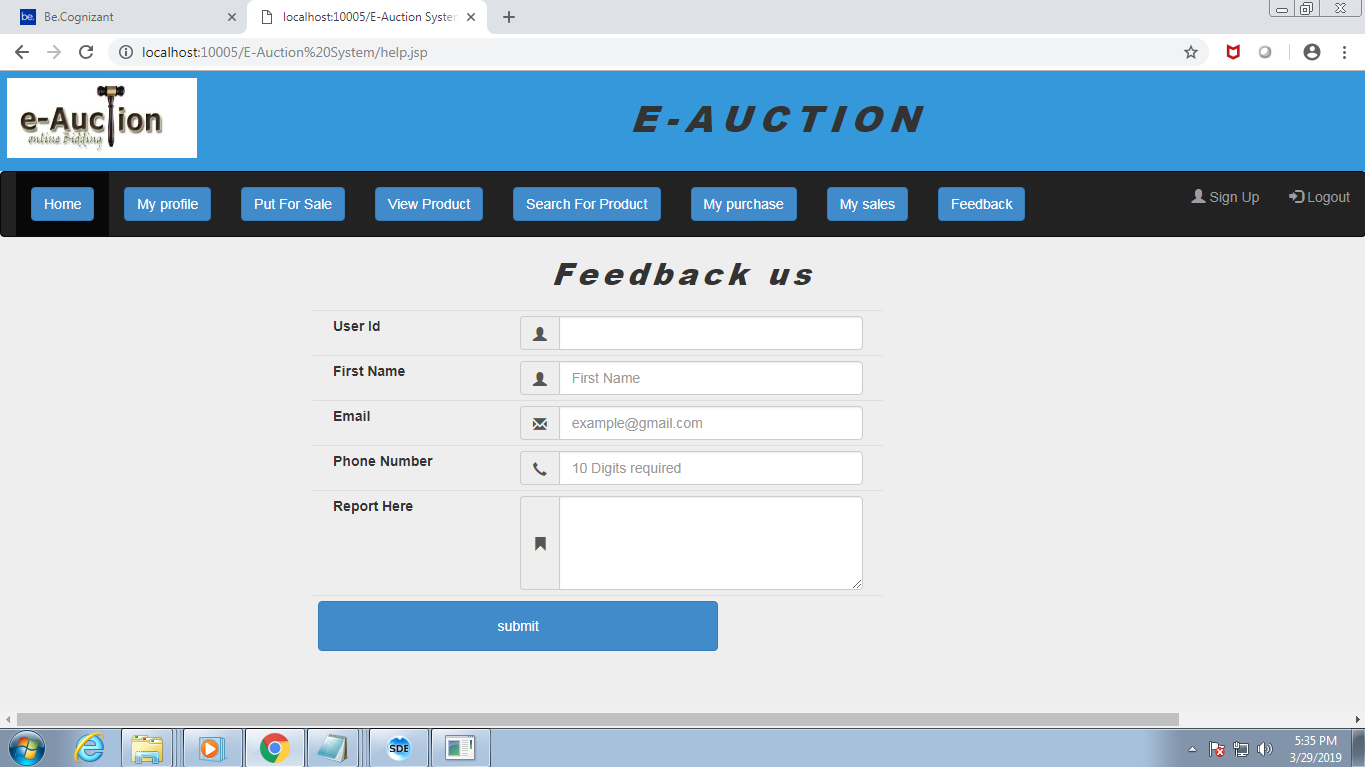
**8.2.6 BIDDING CENTER:**

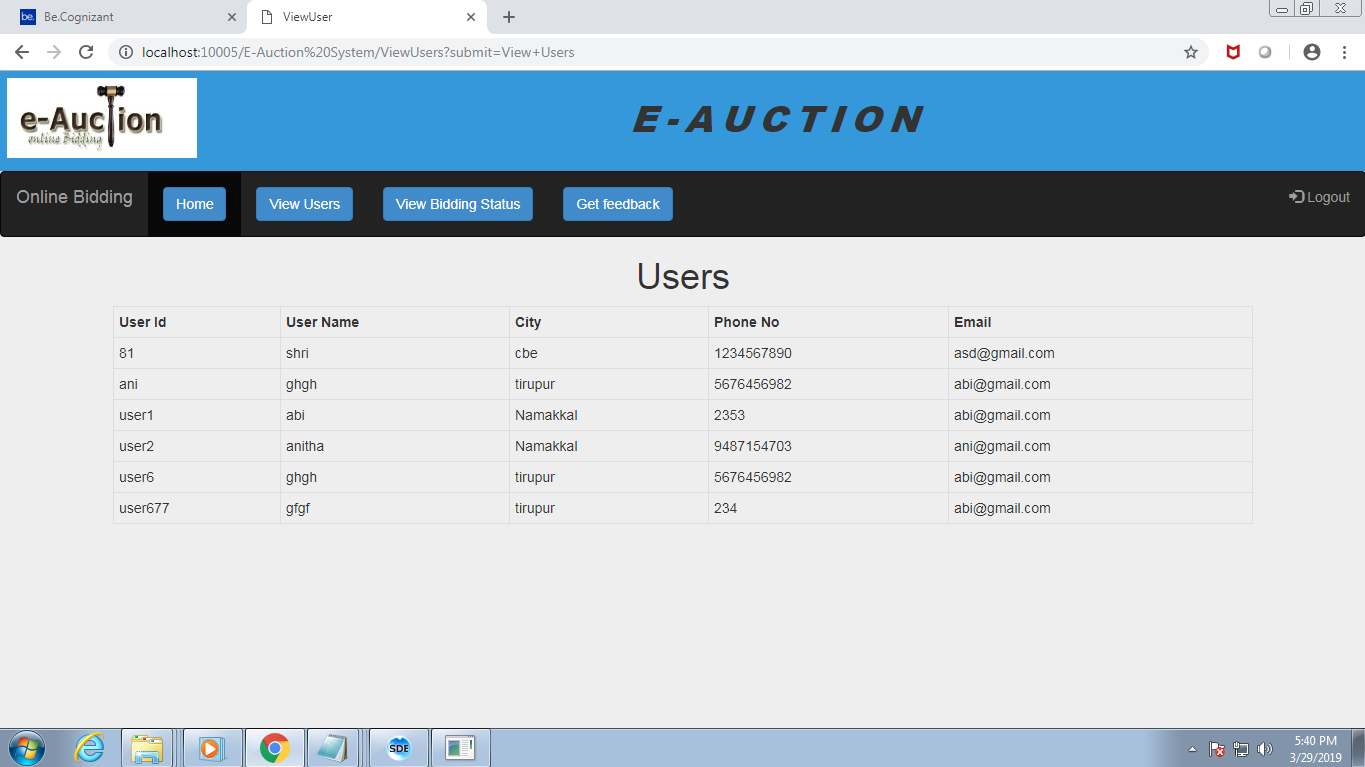
This is the module where multiple users bid for a product and the product is sold for the person who quotes for the highest rate within the end date.

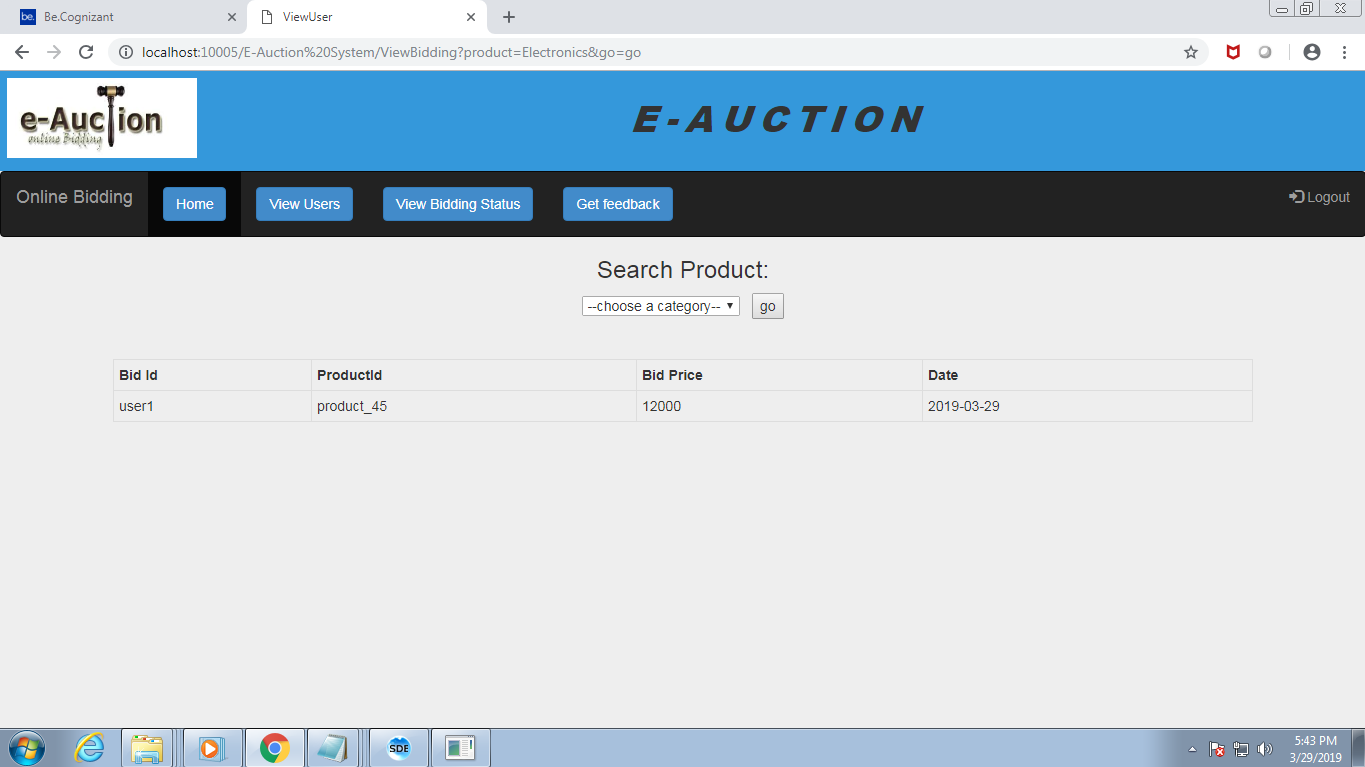


**Fig: 3.6**

****

****

****

****

**CHAPTER 9**

**9. REFRENCES**

Overview of Data Grid view in .net:

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview.rowsdefaultcellstyle?view=netframework-4.7.2>

Uploading files : <https://www.c-sharpcorner.com/uploadfile/aa04e6/autocomplete-combobox-in-datagridview-using-C-Sharp-net-windows-application/>

Generating pdf: <https://bytes.com/topic/c-sharp/answers/442061-autocompletestringcollection-text-box-datagridview-2-0-a>

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | |
|  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |
|  | | |  |

**\**