**ONLINE AUCTION WEBSITE**

A Project Report Submitted to

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY**

In partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

SUBMITTED BY

**VAIBHAV AGARWAL (10BD1A1260)**

**M. ANITH VISHWANATH (10BD1A1258)**

**SADINENI VINEEL (10BD1A1249)**

Under the Guidance of

***(Dr. Siddhartha Ghosh, Professor and HOD, CSE Dept.)***

***(Mr. Abdul Rehman, Assistant Professor, CSE Dept.)***



**DEPARTMENT OF INFORMATION TECHNOLOGY**

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUH)

Narayanaguda, Hyderabad

Year of submission: 2014

**DECLARATION**

We hereby declare that the results embodied in this dissertation entitled “**Online Auction Website**” has been carried out by us together during the academic year 2013-2014 as a partial fulfilment of the award of the B. Tech degree in Information Technology from JNTU-H. We have not copied the same from any source and have not submitted this report to any other university or organization for award of any other degree.

**VAIBHAV AGARWAL (10BD1A1260)**

**M. ANITH VISHWANATH (10BD1A1258)**

**SADINENI VINEEL (10BD1A1249)**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance have been a source of inspiration throughout the course of the project work. We are glad to have the opportunity of expressing our gratitude to all of them.

We are thankful to our Mentor **Dr. Siddhartha Ghosh**, Professor and Head of the Department of Computer Science and Engineering, Keshav Memorial Institute of Technology. We thank him for his support and guidance throughout the project. He is a source of inspiration for innovative ideas and his kind support is well known to all his students and colleagues.

We are thankful to our Guide **Mr. Abdul Rehman**, Assistant Professor of Computer Science and Engineering Dept., Keshav Memorial Institute of Technology, for his support and guidance throughout the project.

We also like to thank the Information Technology Branch Coordinator, **Mr. G. Narender**, Associate Professor, Computer Science and Engineering Dept., Keshav Memorial Institute of Technology. We thank him for providing us timely advice throughout the project work.

We also express our sincere gratitude to the Management and Principal of Keshav Memorial Institute of Technology for their encouragement, facilities provided and support.

Finally, we would like to make a special mention of all our family members and friends who helped us for the successful completion of the seminar report.

**VAIBHAV AGARWAL (10BD1A1260)**

**M. ANITH VISHWANATH (10BD1A1258)**

**SADINENI VINEEL (10BD1A1249)**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that this Project Report is the bonafide work of the following students:

1. **Vaibhav Agarwal 10BD1A1260**
2. **M. Anith Vishwanath 10BD1A1258**
3. **Sadineni Vineel 10BD1A1249**

carried out the Project entitled “**ONLINE AUCTION WEBSITE**” under our supervision.

|  |  |  |
| --- | --- | --- |
| **Internal Guide** |  | **Branch Coordinator** |
| **Mr. Abdul Rehman**  **Assistant Professor, CSE** |  | **Mr. G. Narender**  **Associate Professor, CSE** |
|  |  |  |
|  |  |  |
| **Mentor**  **Dr. Siddhartha Ghosh**  **Professor, CSE** |  | **Head of the Department, IT**  **Dr. Ramakanta Mohanty**  **Professor, IT** |
|  |  |  |

**Submitted for Viva Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner**

**CONTENTS**

**Abstract ...........................................................................................................................................i**

**List of Abbreviations ....................................................................................................................ii**

**List of Figures ..............................................................................................................................iii**

**List of Screens ..............................................................................................................................iv**

**CHAPTER 1**

**1. Introduction ...............................................................................................................................1**

1.1. Purpose of the Project ..........................................................................................................1

1.2. Study report of Important Existing Systems ........................................................................1

1.3. Problems in Existing Systems ..............................................................................................2

1.4. Proposed System and System Architecture .........................................................................3

1.5. Scope of the Project .............................................................................................................4

**CHAPTER 2**

**2. Software Requirement Specifications** **.....................................................................................5**

2.1. Requirements Specification Document ...............................................................................5

2.1.1. Functional Requirements .........................................................................................5

2.1.1.1. User Registration .......................................................................................6

2.1.1.2. User Authentication ...................................................................................6

2.1.1.3. Post Auction ...............................................................................................6

2.1.1.4. View Auctions ...........................................................................................6

2.1.2. Non-Functional Requirements .................................................................................7

2.1.2.1. Performance Requirements ........................................................................7

2.1.2.2. Software Quality Attributes .......................................................................7

2.2 Software Requirements .......................................................................................................8

2.3 Hardware Requirements ......................................................................................................8

2.4 Communications Interfaces ................................................................................................8

**CHAPTER 3**

**3. Technologies/Tools Used for Used for implementation .........................................................9**

3.1 JAVA ....................................................................................................................................9

3.2 J2EE ......................................................................................................................................9

3.3 Database and JDBC ..............................................................................................................9

3.4 Apache Tomcat ...................................................................................................................10

3.5 Eclipse IDE .........................................................................................................................10

3.6 HTML .................................................................................................................................10

3.7 CSS .....................................................................................................................................10

3.8 JavaScript ............................................................................................................................11

3.9 jQuery .................................................................................................................................11

**CHAPTER 4**

**4. Design .......................................................................................................................................12**

4.1 System Architecture ............................................................................................................12

4.1.1 Modules of the System .............................................................................................12

4.1.2 Architecture Diagram ...............................................................................................14

4.2 UML Diagrams ...................................................................................................................15

4.2.1 Class Diagrams ..........................................................................................................15

4.2.1.1 Class Diagram for Overall System ...........................................................15

4.2.1.2 Class Diagram for Registration ................................................................16

4.2.1.3 Class Diagram for Internationalization .....................................................17

4.2.2 Sequence Diagrams ....................................................................................................18

4.2.2.1 Sequence Diagram for Sign in ..................................................................18

4.2.2.2 Sequence Diagram for Post Auction .........................................................19

4.2.2.3 Sequence Diagram for Bidding .................................................................20

4.2.3 Use Case Diagrams ....................................................................................................21

4.2.3.1 Use Case Diagram for System ..................................................................21

4.2.3.2 Use Case Diagram for User ......................................................................22

4.2.3.3 Use Case Diagram for Administrator .......................................................23

4.3 Database Design ...................................................................................................................24

**CHAPTER 5**

**5. Implementation .......................................................................................................................27**

5.1. Classes used .......................................................................................................................27

**CHAPTER 6**

**6. Integration and Testing ..........................................................................................................54**

6.1 Integration ...........................................................................................................................54

6.2 Testing .................................................................................................................................55

6.2.1 Test Cases ................................................................................................................57

**CHAPTER 7**

7.1. Output Screens .......................................................................................................................61

7.2. Future Enhancements .............................................................................................................72

7.3. Conclusion .............................................................................................................................72

7.4. Bibliography ..........................................................................................................................73

**ABSTRACT**

Online auctions are among the most influential e-business applications. Their impact on various trade areas is quite remarkable. This is because online auctions are done mainly through a real-time communication network, viz. the Internet, which basically offers an open marketplace facilitating quicker trade and easy transactions. Also, online auctions break down and remove physical limitations compared to that of traditional auctions such as geography, time, presence, space and a little audience. In other words, online auctions do not put any limits in that any number of audiences can bid on a product in a relatively short period of time, while also offering a fair and just place to trade.

“BidEasy” is an online auction website borrowing the principles of traditional auction houses while also catering to the fundamentals of e-business applications. It mainly aims at making online auctions a safe and easy place to perform various transactions. The whole premise of the website is that it offers a trustable open forum wherein buyers and sellers can negotiate and engage in a quick, hassle-free transaction.

**LIST OF ABBREVIATIONS**

J2SE - Java 2 Standard Edition

J2EE - Java 2 Enterprise Edition

API - Application Programming Interface

JVM - Java Virtual Machine

OS - Operating System

VM - Virtual Machine

SQL - Structural Query Language

ISO - International Organization for Standards

ODBC - Object Database Connectivity

JDBC – Java Database Connectivity

JSON – JavaScript Object Notation

AJAX – Asynchronous JavaScript and XML

XML – Extensible Markup Language

HTTP – HyperText Transfer Protocol

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

JSP – JavaServer Pages

i18n – Internationalization

l10n - Localization

**LIST OF FIGURES**

1. System Architecture ....................................................................................................................3

2. Architecture Diagram ................................................................................................................14

3. Class Diagram (System) ...........................................................................................................15

4. Class Diagram (Registration) ....................................................................................................16

5. Class Diagram (Internationalization) ........................................................................................17

6. Sequence Diagram (Sign in) .....................................................................................................18

7. Sequence Diagram (Post Auction) ............................................................................................19

8. Sequence Diagram (Bidding) ....................................................................................................20

9. Use Case Diagram (System) .....................................................................................................21

10. Use Case Diagram (User) .......................................................................................................22

11. Use Case Diagram (Administrator) ........................................................................................23

**LIST OF TABLES**

1. Auction Table ............................................................................................................................24

2. Bid Table ...................................................................................................................................25

3. Category Table ..........................................................................................................................25

4. Feedback Table .........................................................................................................................25

5. Members Table .........................................................................................................................26

**LIST OF SCREENS**

1. Home/Index Page ......................................................................................................................62

2. Registration Page ......................................................................................................................63

3. Login Page ................................................................................................................................64

4. Post Auction Page .....................................................................................................................64

5. View Auction ............................................................................................................................65

6. Manage Auctions (Admin) .......................................................................................................65

7. Manage Feedback (Admin) .......................................................................................................66

8. About Page ................................................................................................................................66

9. Contact Page .............................................................................................................................67

10. Feedback Page ........................................................................................................................67

11. Chat Page ................................................................................................................................68

12. Help Page ................................................................................................................................69

13. Internationalized Page – Home ...............................................................................................70

14. Internationalized Page – Post Auction ....................................................................................71

**CHAPTER 1**

1. **Introduction**

Auctions represent one of the world’s simplest business models: an item or service is offered for sale to the highest bidder. However, only a small number of people take part in auctions. This is because a conventional auction typically requires travel to a central site where people can sell goods or examine the items that will be put up for bid. Sellers have no guarantee that interested buyers will show up for the auction, and buyers have no guarantee that they will find something they want or need.

Online auctions are among the most influential e-commerce applications. This is because online auctions are done mainly through a real-time communication network, viz. the Internet, which basically offers an open marketplace facilitating quicker trade and easy transactions. Also, online auctions break down and remove physical limitations compared to that of conventional auctions such as geography, time, presence, space and a little audience. In other words, online auctions do not put any limits in that any number of audiences can bid on a product in a relatively short period of time, while also offering a fair and just place to trade.

* 1. **Purpose of the Project**

The main objective of this project is to build an online auction website that borrows the principles of traditional auction houses while also catering to the fundamentals of e-business applications. It mainly aims at making online auctions a safe and easy place to perform various transactions. The whole premise of the website is that it offers a trustable open forum wherein buyers and sellers can negotiate and engage in a quick, hassle-free transaction.

* 1. **Study Report of Important Existing Systems**

In the past few years, the electronic marketplace has witnessed an exponential growth in worth, size, and usability. Thus, there has been a massive expansion of online commercial services offering various facilities to the user.

The services can be categorized into two business models: a Customer-to-Customer (C2C) model, wherein customers can act as both bidders and sellers of their products, and a Business-to-Customer (B2C) model, in which the website itself is responsible for hosting the items/products. However, there are no differences with regard to the implementation of the core functionality in either of the two models.

The auctions are running in real time, and they may vary. Typical auction times are 1-3 hours, consisting of a main part and an extension part. The extension part comes in when a bidder bids on the product when it is nearing its auction time, allowing other bidders to react.

Only the seller can correlate the bids to the bidders during an auction. Bidders appear to each other anonymously and once the auction time expires (either because the time expired or because a bidder has won), a final report of the auction result is provided to all the participants.

A login system has been found to be imperative on all of the major service websites. Without login or registration, the user can simply only view various products offered on the website.

* 1. **Problems in Existing Systems**

Some drawbacks found in existing systems have been enlisted below:

* In many systems, there is a lack of a good Graphical User Interface. They have many links that might confuse the user and give the website a burdened appearance.
* Chat feature is not implemented in many systems.
* If there are many bids at the same time on a product, then some of the bids might be lost because the server might not be able to handle all bids.
* Internationalization/Localization is not supported or is only supported partially.
  1. **Proposed System and System Architecture**

Our system follows a Customer-to-Customer (C2C) business model in which customers have the liberty to bid or sell at any time only if they are authenticated. A chat feature is implemented with the help of an Internet Relay Chat (IRC) chat room where users can acquire support. Complete internationalization/localization support to all the modules of the website, and a Help section that guides users having difficulty navigating through the website.

**System Architecture:**

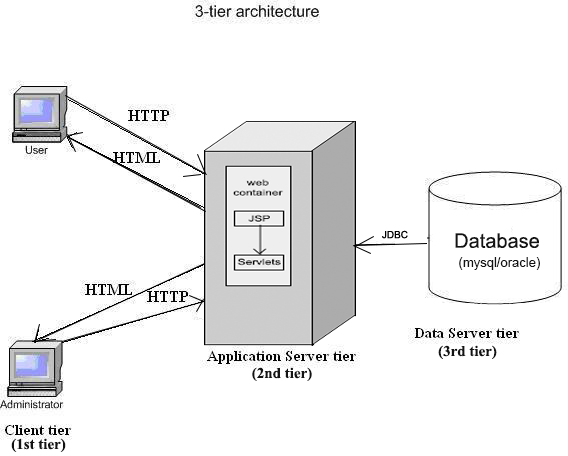
****

Figure 1: System Architecture

In the above architecture, there are three main elements:

* The client tier, that is responsible for the presentation of data, receiving user elements and controlling the user interface.
* The **application server tier,** that is responsible for the business logic of the system. In fact, business objects that implement the business rules “reside” here and are available to the client tier. For the project, we used Apache Tomcat as the application server that consists of a web container that can handle both JSPs and Servlets.
* The **data server tier,** that is responsible for data storage. As data server, we used MySQL, an open-source database that is lightweight and easy to setup.

**1.5. Scope of the Project**

The scope of this project first establishes an entry threshold by identifying the users and their associated roles. The administrator manages the website; and various users can gain entry through registration and verification process so as to avail bidding and selling services. In fact, the core module of the website is based on user registration. Registered users have to “activate/authenticate” their account after registering. This is done by users entering their activation code that is sent to their e-mail addresses.

Auctions have a name, a unique identification number and a photo (of the related item) that the seller can upload, the start and end dates. Bidders cannot bid on an auction once the auction time expires. Moreover, administrators have the possibility to accept or refuse auctions proposed by users.

The system is realized with a three-tier architecture: 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.

**CHAPTER 2**

1. **Software Requirements Specification (SRS)**
   1. **Requirements Specification Document**

A Software requirements specification (SRS), a requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. This document is very comprehensive document and contains all the user requirements and analysis diagrams.

The requirements are broadly classified into two types:

* Functional requirements
* Non-functional requirements
  + 1. **Functional Requirements** 
       1. **User Registration**
          1. Description and Priority:

Allows users to become members of the system and enables them to participate in community auctions. The registration module is imperative and guests can avail services only once registered and fully authenticated.

* + - * 1. Stimulus/Response Sequences:

1. Users click on “Register” on the top right-hand corner of the page to get to the registration form and after following the instructions, fill in the required information, thereby creating their own profile.
2. Client-side validation is performed for checking provided information.
3. Users after successfully registering are sent an e-mail thereby confirming their registration.
   * + 1. **User Authentication**
          1. Description and Priority:

Allows users to be authenticated which qualify them to participate in the services the community offers.

* + - * 1. Stimulus/Response Sequences:

A unique confirmation/activation code is sent to the users’ e-mail addresses, which when entered successfully authenticates their profile.

* + - 1. **Post Auction**
         1. Description and Priority:

Allows users to sell items up for bidding. Only authenticated users can perform this activity.

* + - * 1. Stimulus/Response Sequences:

1. Authenticated users click on “Post Auction” on the top and fill the post auction form with the required information like Product name, Product Image, Description, Reserved and Start Price, and the Auction Start and End dates.
2. After filling the information successfully, a notification mail is sent to the seller informing the seller that the item has been posted on the website. The product is given a unique Product ID and is authorized for auctioning.
   * + 1. **View Auctions**
          1. Description and Priority:

Allows authenticated users to browse on the products currently active for auctioning in the system. Also allows users to bid on the products available in the system.

* + - * 1. Stimulus/Response Sequences:

1. Authenticated users click on “View Auctions” on the top and can view the products currently available for auctioning.
2. A “Bid” button is placed next to the items so that prospective buyers can bid on a particular item.
   * 1. **Non Functional Requirements**

**2.1.2.1 Performance Requirements**

A central issue pertaining to hardware implementation is that if large amounts of traffic (visitors/guests) are trying to access the site simultaneously, say, in order to bid on a popular auction at closing time, then the server may not be able to handle the traffic thus inadvertently taking down the website. Another implementation issue arises when the Tomcat server that hosts the JSP function crashes or fails.

**2.1.2.2 Software Quality Attributes**

Usability: The software is easy to use as it does not require any prior training. Moreover, a help page has been provided for users who are new to the system and who need further assistance with respect to the functionality of the system.

Portability: The website is portable and can be used across a range of Web Browsers.

Scalability: The website needs to address the scalability issue, that is, when there is huge traffic at a single point in time, the server should be able to handle the load adequately.

* 1. **Software Requirements**

The following are the software requirements:

* **Client on Internet:** Web Browser (Google Chrome, Firefox, Safari), Operating System (Windows xp minimum)
* **Application Server:** Apache Tomcat
* **Database Server:** MySQL 5.1
* **Network:** Internet
* **Development Tools:** Sublime Text Editor, Eclipse IDE (Juno), Visual Paradigm, MySQL Workbench, Apache Tomcat
  1. **Hardware Requirements**

The website is accessible from any operating system using a web browser. No special hardware is required by the end-user. The clients’ browser must be W3C XHTML compatible. Communication between the users and the auction site will be via HTTP/HTTPS communication using TCP/IP port number 80. If an error occurs during a request, the user should receive a clear error message.

* **Processor:** Intel Pentium 4 or above
* **RAM:** 512MB or above
* **Hard Disk Drive:** 80GB minimum
* **Monitor:** SVGA monitor that supports a minimum resolution of 1024x768 pixels
* **Keyboard and Mouse:** Standard – USB or PS/2 compliant
* **Modem:** 256 kbps or ADSL Broadband
  1. **Communications Interfaces**
* Client on Internet will be using HTTP/HTTPS protocol.
* Client on Intranet will be using TCP/IP protocol.
* A web browser such as Google Chrome or equivalent.

**CHAPTER 3**

**3. Technologies used for implementation**

**3.1 JAVA**

Java is an object-oriented programming language developed by Sun Microsystems a company best known for its high end UNIX workstations. Java language was designed to be small, simple, and portable across platforms, operating systems, both at the source and at the binary level, which means that Java programs (applet and application) can run on any machine that has the Java virtual machine (JVM) installed. The language derives much of its [syntax](http://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](http://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.  Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications.

**3.2 J2EE**

Java Platform, Enterprise Editionor Java EEis a widely used platform for server programming in the Java programming language. The Java platform (Enterprise Edition) differs from the Java Standard Edition Platform (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server. Java EE includes several API specifications, such as JDBC, RMI, e-mail, JMS, web services, XML, etc., and defines how to coordinate them. Java EE also features some specifications unique to Java EE for components. These include Enterprise JavaBeans, Connectors, servlets, JavaServer Pages (JSPs) and several web service technologies. This allows developers to create enterprise applications that are portable and scalable, and that integrate with legacy technologies. A Java EE application server can handle transactions, security, scalability, concurrency and management of the components it is deploying, in order to enable developers to concentrate more on the business logic of the components rather than on infrastructure and integration tasks.

**3.3 Database and JDBC**

JDBC is a Java-based data access technology (Java Standard Edition platform) from Oracle Corporation. This technology is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the JVM host environment.

***MySQL*** is the world's most widely used open-source relational database management system (RDBMS). MySQL is a popular choice of database for use in web applications. This is because MySQL is open-source software, has low machine requirements and is easy to setup.

**3.4 Apache Tomcat**

Apache Tomcat (or simply Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation. Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run in. It is bundled with Catalina (a servlet container), Coyote (a HTTP connector) and Jasper (a [JSP engine](http://en.wikipedia.org/wiki/JSP_engine)). It implements the Servlet 3.0 and JSP 2.2 specifications.

**3.6 Eclipse IDE**

Eclipse is an integrated development environment (IDE) written mostly in Java. It can be used to develop applications. Development environments include the Eclipse Java development tools (JDT) for Java. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Eclipse supports development for Tomcat and many other servers and is often capable of installing the required server (for development) directly from the IDE. It supports remote debugging, allowing the user to watch variables and step through the code of an application that is running on the attached server.

**3.7 HTML**

HTML or HyperText Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

**3.8 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. CSS is most often used to style web pages and interfaces written in HTML. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table-less web design).

**3.9 JavaScript**

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming and the creation of desktop and mobile applications.

**3.10 jQuery**

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. Used by over 80% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is free, open source software, licensed under the MIT License. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

The jQuery library is a single JavaScript file, containing all of its common DOM, event, effects, and Ajax functions. It can be included within a web page by linking to a local copy, or to one of the many copies available from public servers.

**CHAPTER 4**

**4. Design**

**4.1 System Architecture**

**4.1.1 Modules of the System**

**Module 1 – Registration**

In this module, users can register an account in the website. Users click on “Register” on the top right-hand corner of the screen to get to the registration form and after following the instructions, fill in the required information, thereby creating their own profile. Once filling in the information, users are sent a confirmation mail to their e-mail addresses and upon entering the activation code are granted authentication to participate in the website services.

**Module 2 – Sign in**

This module allows users to login to the system. The prerequisite for this module is that the user must have a valid, authenticated account by first registering in the website. Once registered, they can go to the login page by clicking on the “Sign in” link on the top right-hand corner of the screen, and enter their username and passwords to gain access to the website.

**Module 3 – Post Auctions**

In this module, users can post their items on the website. The users have to be logged in to be able to perform this action. If the users are not logged in, they are redirected to the Sign in page. Users have to fill a form and once complete, a notification mail goes to the users’ e-mail address stating that their product is up for auctioning in the system.

**Module 4 – View Auctions/Bid**

Allows authenticated users to view the list of items present in the system available for open bidding. Here, users can bid on various items present in the auction lot. Sellers while posting their item decide on the start price and the reserved price, i.e., the price below which the item cannot be sold. Bidders bid on a product currently available in the system by clicking on the Bid Button and typing in their price. A notification mail goes to the seller’s e-mail stating that a bidder has bid on their product. Additionally, an e-mail also goes to all the bidders (if any) who have bid on the product.

**Module 5 – Chat**

The Chat module aims primarily at providing help to users. The module is hosted on IRC and is provided with instructions for users unfamiliar with the chat room environment.

**Module 6 – Feedback**

This module allows registered users to give feedback and complaints about the system. Users’ feedbacks are stored in a database, and admins may respond to each feedback. An e-mail goes to the user when an admin has responded to a user’s feedback.

**Module 7 – Change Language**

This module allows users to change their language. This is done via a simple drop-down menu at the footer of every page. Once the user selects a language, the language change is reflected on all the pages.

**Module 8 – Administration**

Administrators have additional permissions or accessibility rights in the operation of the website. When an administrator logs in, he/she can perform extra duties such as managing, deleting and reviewing auctions, and also managing the feedback queries and complaints.

**Module 9 – Search**

Allows registered users to search for a specific item by categories.

**Module 10 – Help and FAQs Section**

An intuitive help section that guides new users with their most frequently asked questions on how to use the website. Also includes a video highlighting all of the core functionality.

* + 1. **Architecture Diagram**

An architecture diagramis a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem. The diagram below shows the architecture diagram of the online auction system:

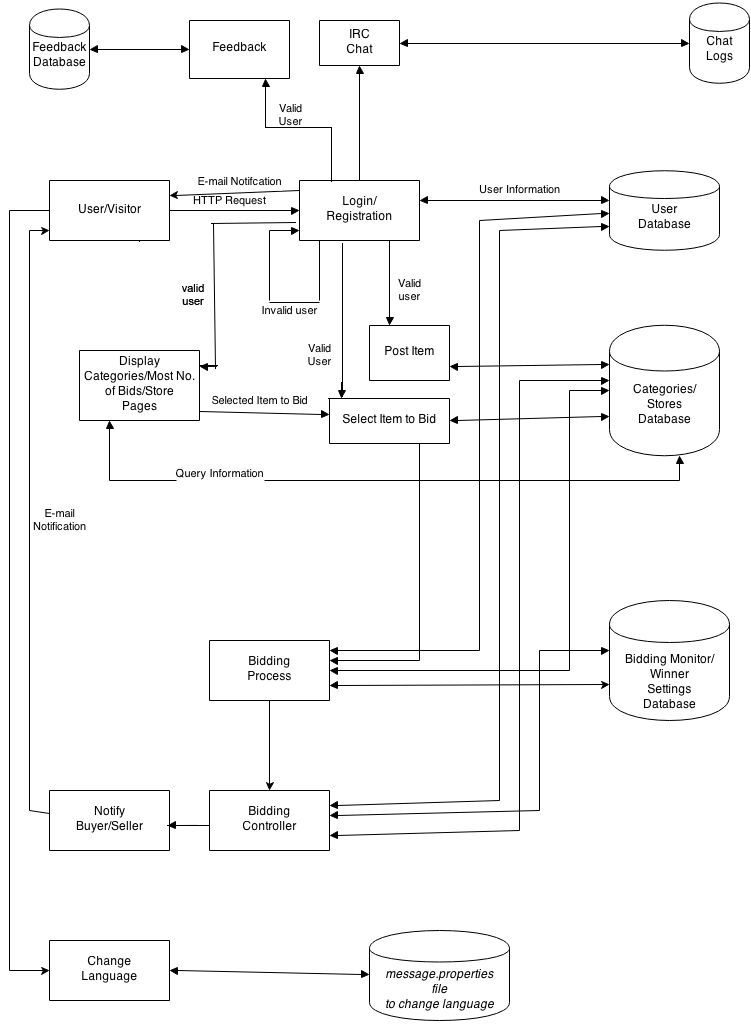


Figure 4.1: Architecture Diagram

**4.2 UML Diagrams**

**4.2.1 Class Diagrams**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

**4.2.1.1 Class Diagram for Overall System**

The class diagram highlighting the various classes, methods, attributes and their relationships is shown below:

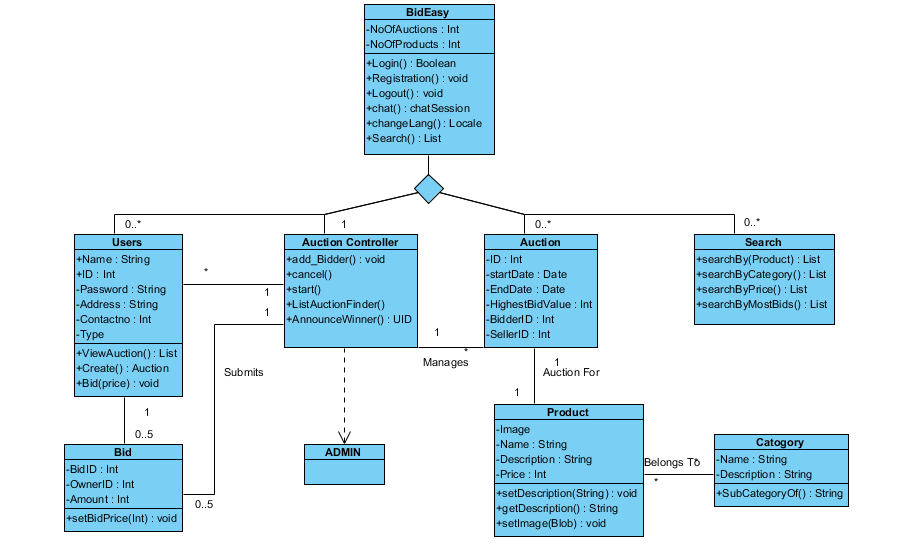


Figure 4.2: Class Diagram describing the overall System

**4.2.1.2 Class Diagram for Registration**

The class diagram for registration is shown below:

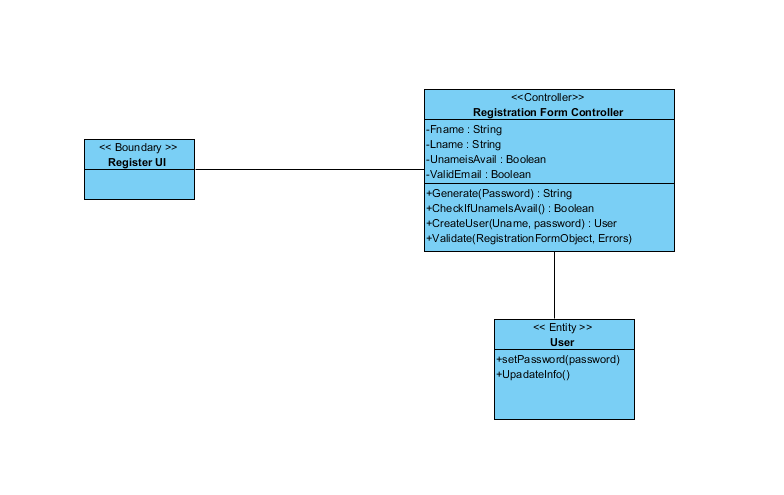


Figure 4.3: Class Diagram for Registration

**4.2.1.3 Class Diagram for Internationalization**

The class diagram for internationalization is shown below:

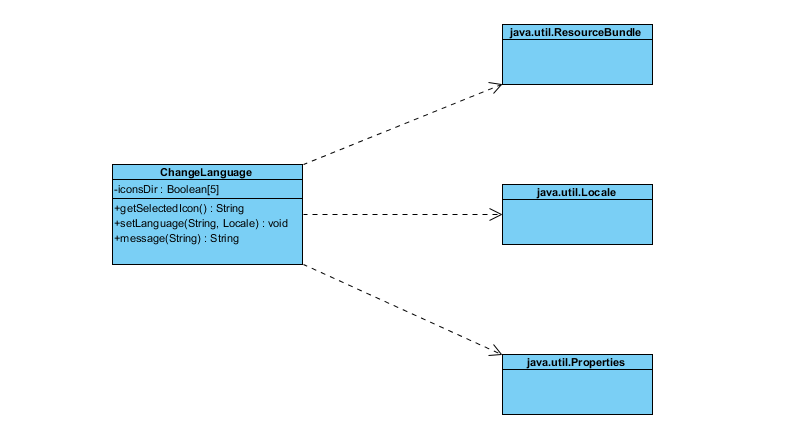


Figure 4.4: Class Diagram for Internationalization

**4.2.2 Sequence Diagrams**

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.2.2.1 Sequence Diagram for Sign in**

The sequence diagram for sign in is shown below:

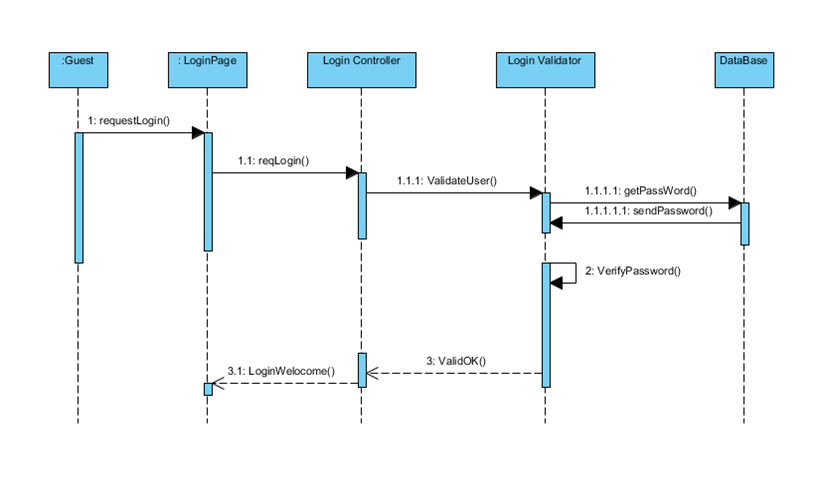


Figure 4.5: Sequence Diagram for Sign in

**4.2.2.2 Sequence Diagram for Post Auction**

The sequence diagram for posting an auction is shown below:

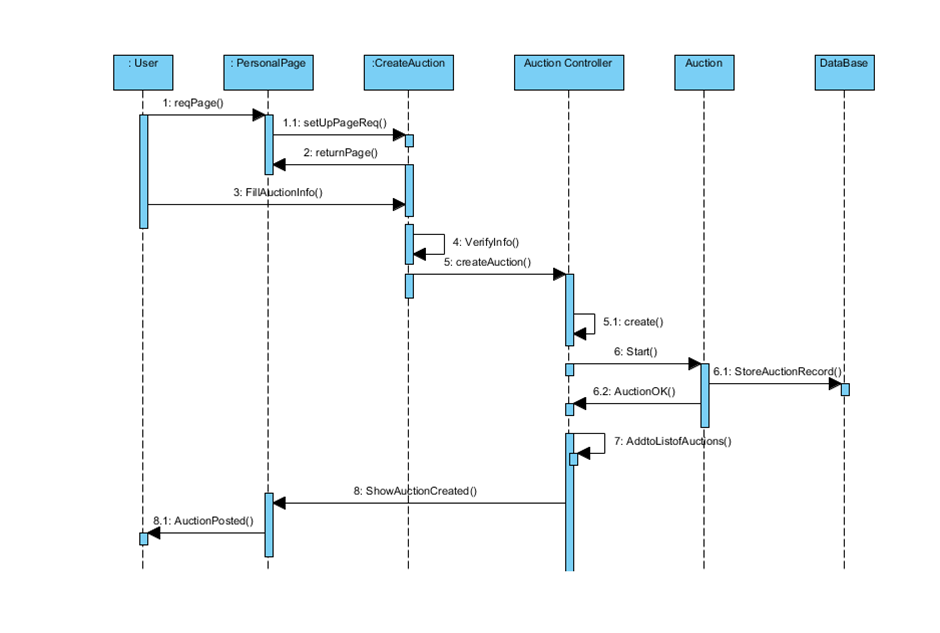


Figure 4.6: Sequence Diagram for Post Auction

**4.2.2.3 Sequence Diagram for Bidding**

The sequence diagram for bidding is shown below:

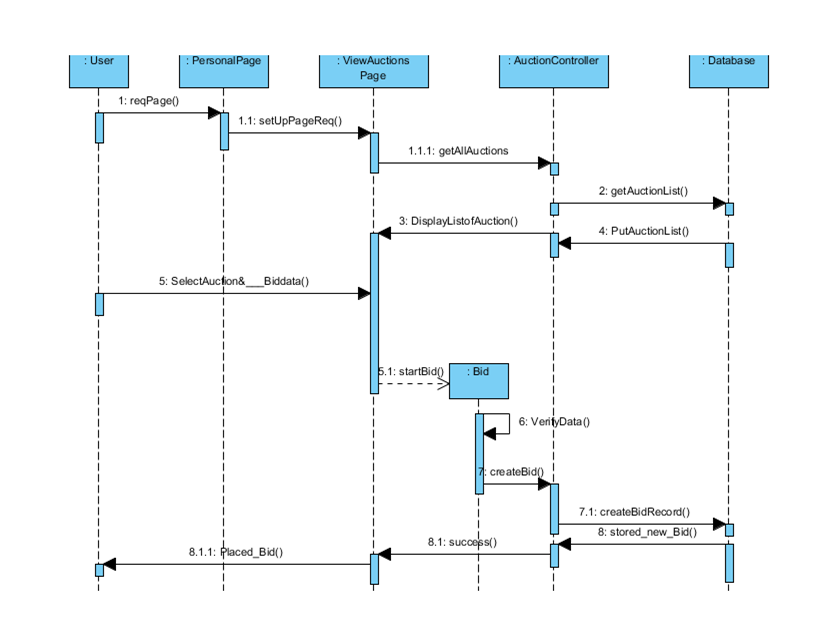


Figure 4.7: Sequence Diagram for Bidding

**4.2.3 USE CASE DIAGRAM**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

**4.2.3.1 Use Case Diagram for system**

The use case diagram depicting the overall system is shown below:

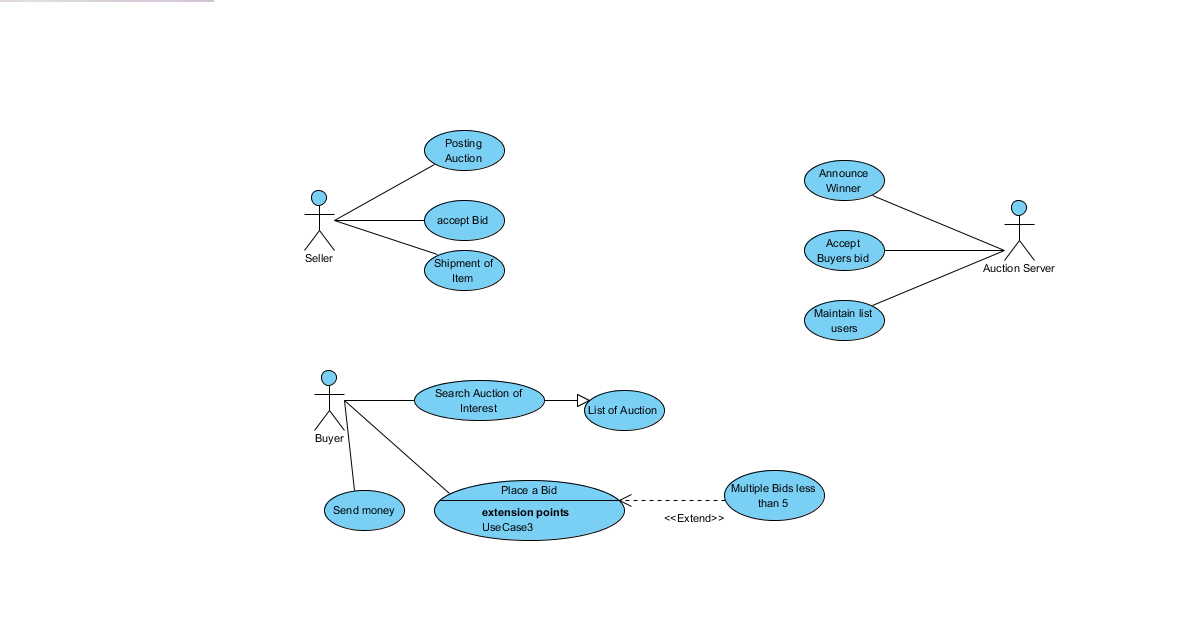


Figure 4.8: Use Case Diagram for overall Auction system

**4.2.3.2 Use Case Diagram for User**

The use case diagram representing a user is shown below:

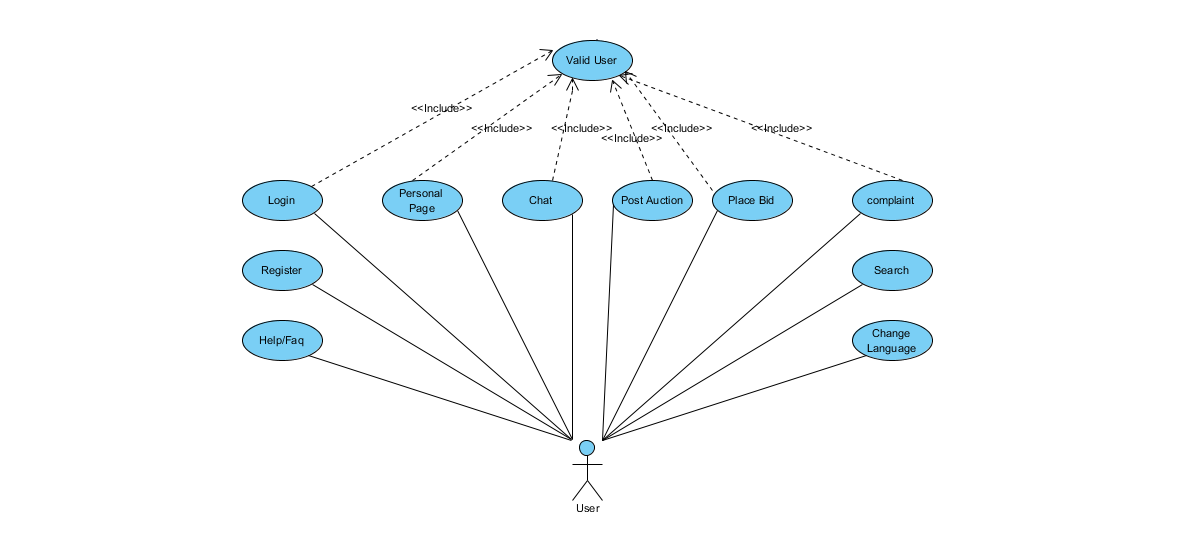


Figure 4.9: Use Case Diagram for User

**4.2.3.3 Use Case Diagram for Administrator**

The use case diagram representing an administrator is shown below:

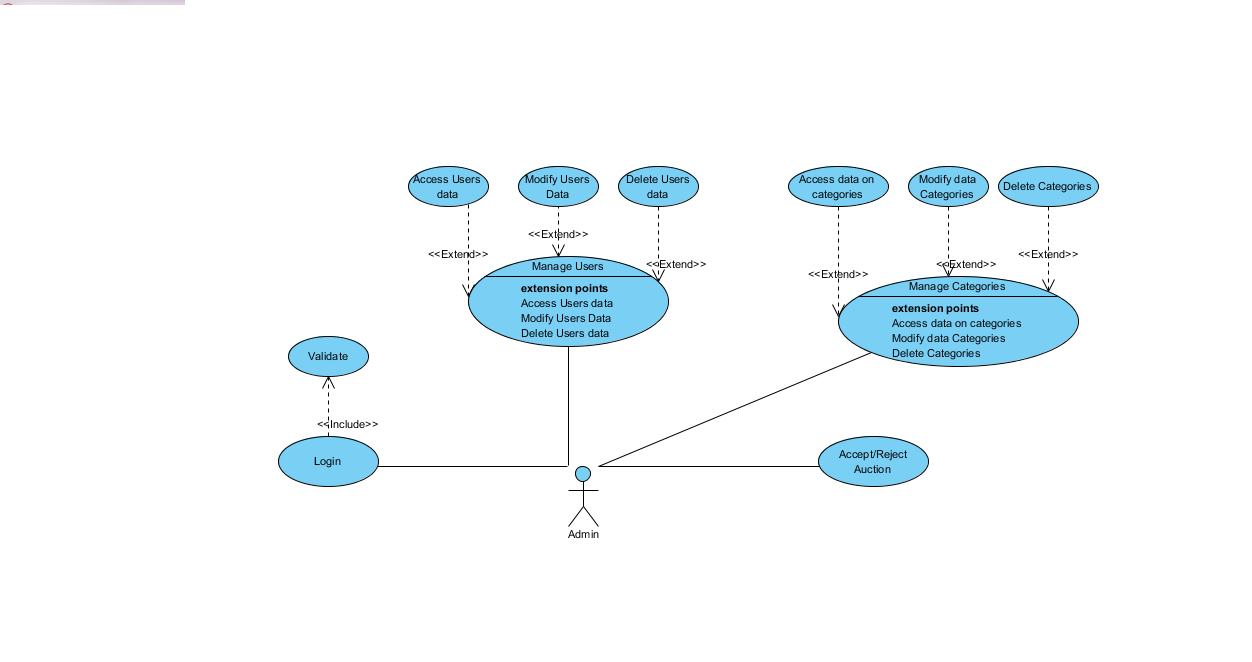


Figure 4.10: Use Case Diagram for Administrator

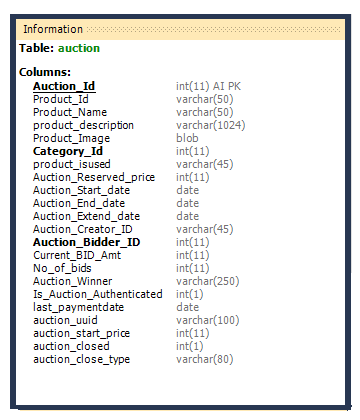
**4.2 Database Design**

MySQL is used to manage the database and JDBC is used to provide Database services. MySQL is used because it is easy to setup and is relatively easy to maintain.

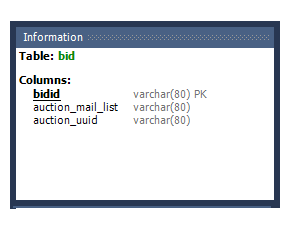
**4.2.1 Database Tables**

The following Database tables have been used in the development of the website:

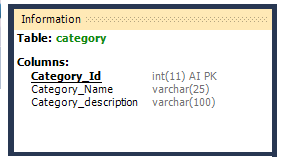
**4.2.1.1 Auctions Table**

****

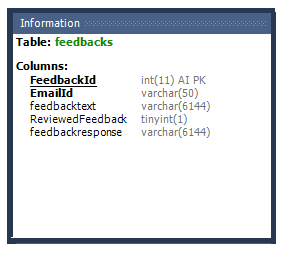
**4.2.1.2 Bid Table**



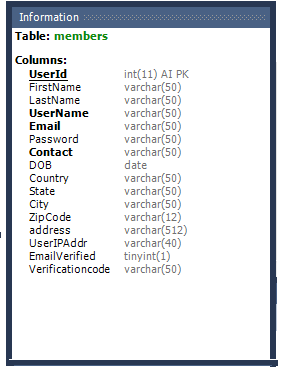
**4.2.1.3 Category Table**

****

**4.2.1.4 Feedback Table**

****

**4.2.1.5 Members Table**

****

**CHAPTER 5**

**5. Implementation**

**5.1 Classes Used**

**Bid:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class Bid extends HttpServlet

{

Connection con=null;

PreparedStatement ps=null;

ResultSet rs1=null;

ResultSet rs2=null;

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

String auction\_uuid=request.getParameter("auction\_uuid");

int bidder\_id=Integer.parseInt(request.getParameter("bidder\_userId"));

int bid\_amt=Integer.parseInt(request.getParameter("bid\_amt"));

String bidder\_email=(String)request.getSession(false).getAttribute("emailId");

String bid\_uuid=UUID.randomUUID().toString().replaceAll("-", "");

String product\_info=request.getParameter("product\_info");

String author\_email=request.getParameter("author\_email");

List<String>mail\_list = new ArrayList<String>();

String query;

boolean is\_new\_bidder=true;

int tot\_bids=0;

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy","root", "root");

query="select no\_of\_bids from auction where auction\_uuid=?";

ps= con.prepareStatement(query);

ps.setString(1,auction\_uuid );

rs1=ps.executeQuery();

if(rs1!=null && rs1.next() )

{

tot\_bids=Integer.parseInt(rs1.getString("no\_of\_bids"));

}

query="update auction set auction\_bidder\_id=?, no\_of\_bids=?,current\_bid\_amt=?,auction\_start\_price=? where auction\_uuid=?";

ps= con.prepareStatement(query);

ps.setInt(1,bidder\_id);

ps.setInt(2,tot\_bids);

ps.setInt(3,bid\_amt);

ps.setInt(4,bid\_amt);

ps.setString(5,auction\_uuid);

int i = ps.executeUpdate();

query="select distinct auction\_mail\_list from bid where auction\_uuid=?";

ps= con.prepareStatement(query);

ps.setString(1,auction\_uuid );

rs2=ps.executeQuery();

while(rs2!=null && rs2.next())

{

mail\_list.add(rs2.getString("auction\_mail\_list"));

}

for (String eachmail : mail\_list )

{

if (eachmail.contains(bidder\_email))

{

is\_new\_bidder=false;

}

else

{

is\_new\_bidder=true;

}

}

String recipients[]=mail\_list.toArray(new String[mail\_list.size()]);

int j=0;

if(is\_new\_bidder)

{

j =insert\_record\_in\_bid\_table(bid\_uuid,bidder\_email,auction\_uuid);

}

else

{

query="delete from bid where auction\_mail\_list=?";

ps= con.prepareStatement(query);

ps.setString(1,bidder\_email);

int k=ps.executeUpdate();

if(k>0)

j= insert\_record\_in\_bid\_table(bid\_uuid,bidder\_email,auction\_uuid);

}

if(i>0 && j>0)

{

try

{

new MailSend().SendMail(author\_email,"Bid Offers to Your Product","A user have shown Interest on your Product-"+product\_info+"\n your Product's Current Bid Amount is"+bid\_amt+"\n Bidder Email id is"+ bidder\_email);

new MailSend().SendMail(bidder\_email,"Thanks for Bidding! ","You Have successfully bid upon Product "+product\_info+" \n your bid id is "+bid\_uuid+"\n Please stay tuned for updates.");

new MailSend().SendBulkMail(recipients,"Auction Notification","Most Recent Bid on product "+product\_info+" is at price "+bid\_amt+"\n Bid more n more to Win! \n Please stay tuned for updates.");

}

catch (Exception e){}

try{

ServletContext context= getServletContext();

request.setAttribute("Bid\_msg", "Successful Bid!");

RequestDispatcher rd= context.getRequestDispatcher("/viewad.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

else

{

try{

ServletContext context= getServletContext();

request.setAttribute("Bid\_error\_msg", "Bid UnSuccessful! Try again.");

RequestDispatcher rd= context.getRequestDispatcher("/viewad.jsp");

rd.forward(request, response);

}

catch(Exception e){}

}

}

catch (SQLException e)

{

e.printStackTrace();

try{

ServletContext context= getServletContext();

request.setAttribute("Bid\_error\_msg", "Bid UnSuccessful! Try again.");

RequestDispatcher rd= context.getRequestDispatcher("/viewad.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (ps != null) {

try {

ps.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (rs1 != null) {

try {

rs1.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (rs2 != null) {

try {

rs2.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

public int insert\_record\_in\_bid\_table(String bid\_id,String bider\_email,String auction\_uid)

{

String query="insert into bid (bidid,auction\_mail\_list,auction\_uuid) values (?,?,?)";

try{

ps= con.prepareStatement(query);

ps.setString(1,bid\_id);

ps.setString(2,bider\_email);

ps.setString(3,auction\_uid);

int res=ps.executeUpdate();

return res;

}

catch (SQLException e)

{

e.printStackTrace();

return 0;

}

}

}

**Post Auction:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class PostAuction extends HttpServlet

{

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException,ServletException

{

Connection con=null;

Statement st=null;

try

{

String pimage=request.getParameter("pimage");

String pname=request.getParameter("pname");

String category=request.getParameter("category");

String pdescription=request.getParameter("description");

String start\_date=request.getParameter("start\_date");

String end\_date=request.getParameter("end\_date");

String isUsed=request.getParameter("p\_use");

String res\_price=request.getParameter("reserved\_price");

String start\_price=request.getParameter("start\_price");

String pid=UUID.randomUUID().toString().replaceAll("-", "");

String aucid=UUID.randomUUID().toString().replaceAll("-", "");

String auth\_email\_id=(String)request.getSession(false).getAttribute("emailId");

int catid=Integer.parseInt(category);

int st\_price=Integer.parseInt(start\_price);

int rsv\_price=Integer.parseInt(res\_price);

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy","root", "root");

String query;

query="insert into auction(product\_id,product\_name,product\_image,product\_description,category\_id,product\_isUsed,auction\_reserved\_price,auction\_start\_date,auction\_end\_date,auction\_creator\_id,Is\_auction\_authenticated,auction\_uuid,auction\_start\_price,auction\_closed,no\_of\_bids) "

+ "values ('"+pid+"','"+ pname +"','"+pimage+"','"+

pdescription+"','"+catid+"','"+isUsed+"',"+rsv\_price+",'"+start\_date+"','"+end\_date+"','"+

auth\_email\_id+"','"+0+"','"+aucid+"','"+st\_price+"',"+0+","+0+")";

st = con.createStatement();

int i = st.executeUpdate(query);

if(i>0)

{

try

{

new MailSend().SendMail(auth\_email\_id,"Thanks","Thank you for posting add on our website your auction uuid is "+aucid);

}

catch (Exception e){}

try{

ServletContext context= getServletContext();

RequestDispatcher rd= context.getRequestDispatcher("/posted\_success.jsp");

rd.forward(request, response);

}

catch(Exception e){}

}

else

{

try{

ServletContext context= getServletContext();

request.setAttribute("postingAd\_error\_msg", "Error in Posting Add.Plz Try Again!");

RequestDispatcher rd= context.getRequestDispatcher("/sell.jsp");

rd.forward(request, response);

}

catch(Exception e){}

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

try{

ServletContext context= getServletContext();

request.setAttribute("postingAd\_error\_msg", "Error in Posting Add.Plz Try Again!");

RequestDispatcher rd= context.getRequestDispatcher("/sell.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (st != null) {

try {

st.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException,ServletException

{

doGet(request,response);

}

}

**Registration:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class Registration extends HttpServlet

{

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

Statement st=null;

Connection con=null;

String user = request.getParameter("uname");

String pwd = request.getParameter("pass");

String fname = request.getParameter("fname");

String lname = request.getParameter("lname");

String email = request.getParameter("email");

String phone= request.getParameter("phone");

String birthdate= request.getParameter("dob");

String country = request.getParameter("country");

String state= request.getParameter("state");

String city = request.getParameter("city");

String zip = request.getParameter("zip");

String Houseaddress = request.getParameter("address");

String ipaddress=request.getRemoteAddr();

RandomPasswordGenerator rpg=new RandomPasswordGenerator();

char verifycode[]=rpg.generatePswd();

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy",

"root", "root");

st = con.createStatement();

int i = st.executeUpdate("insert into members(FirstName,LastName,UserName,Email,Password,Contact,DOB,Country,State,City,Zipcode,Address,UserIPAddr,EmailVerified,Verificationcode) " + "values ('"+fname+"','"+ lname +"','"+user+"','"+email+"','"+pwd+"','"+phone+"','"+birthdate+"','"+country+"','"+state+"','"+city+"','"+zip+"','"+Houseaddress+"','"+ipaddress+"','"+0+"','"+verifycode+"')");

if(i>0)

{

try

{

new MailSend().SendMail(email,"Welcome to BidEsay","Hi,"+fname+"Your Verification Code is "+verifycode);

}

catch (Exception e)

{

}

try{

ServletContext context= getServletContext();

RequestDispatcher rd= context.getRequestDispatcher("/welcome.jsp");

rd.forward(request, response);

}

catch(Exception e){}

}

else

{

try{

ServletContext context= getServletContext();

request.setAttribute("Reg\_error\_msg", "Error in Registration.Plz Try Again!");

RequestDispatcher rd= context.getRequestDispatcher("/register.jsp");

rd.forward(request, response);

}

catch(Exception e){}

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

try{

ServletContext context= getServletContext();

request.setAttribute("Reg\_error\_msg", "Error in Registration.Plz Try Again!");

RequestDispatcher rd= context.getRequestDispatcher("/register.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (st != null) {

try {

st.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}}

**Dologin:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

public class Dologin extends HttpServlet

{

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

Connection conn = null;

ResultSet rsdoLogin = null;

PreparedStatement psdoLogin1=null;

ResultSet rsdoLogin1 = null;

PreparedStatement psdoLogin=null;

try

{

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy","root", "root");

String sUserID=request.getParameter("login\_username");

String sPassword=request.getParameter("login\_password");

HttpSession session = request.getSession(true);

String sqlOption="SELECT \* FROM members where"

+" UserName=? and Password=(?) and EmailVerified=1 ;";

String sqlOption1="SELECT \* FROM members where"

+" UserName=? and Password=(?) ;";

psdoLogin=conn.prepareStatement(sqlOption);

psdoLogin.setString(1,sUserID);

psdoLogin.setString(2,sPassword);

rsdoLogin=psdoLogin.executeQuery();

psdoLogin1=conn.prepareStatement(sqlOption1);

psdoLogin1.setString(1,sUserID);

psdoLogin1.setString(2,sPassword);

rsdoLogin1=psdoLogin1.executeQuery();

if(rsdoLogin!=null && rsdoLogin.next())

{

String FullName=rsdoLogin.getString("FirstName")+" "+rsdoLogin.getString("LastName");

session.setAttribute("Username",rsdoLogin.getString("UserName"));

session.setAttribute("fullname",FullName);

session.setAttribute("userId",rsdoLogin.getString("userid"));

session.setAttribute("emailId",rsdoLogin.getString("Email"));

try{

response.sendRedirect("index.jsp");

return;

}

catch(Exception e){}

}

else

{

if(rsdoLogin1!=null && rsdoLogin1.next())

{

try{

ServletContext context= getServletContext();

request.getSession().setAttribute("login\_error\_msg", "You are not verified user.Plz verify!");

response.sendRedirect("welcome.jsp");

return;

}

catch(Exception e){}

}

else{

try{

ServletContext context= getServletContext();

request.getSession().setAttribute("login\_error\_msg", "UserName and Password Do Not Match.Plz Try Again!");

response.sendRedirect("login.jsp");

return;

}

catch(Exception e){}

}

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (rsdoLogin!= null) {

try {

rsdoLogin.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (rsdoLogin1!= null) {

try {

rsdoLogin1.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (psdoLogin!= null) {

try {

psdoLogin.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (psdoLogin1!= null) {

try {

psdoLogin1.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (conn != null) {

try {

conn.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

}

**Feedback:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class Feedback extends HttpServlet

{

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

Statement st=null;

Connection con=null;

String email = request.getParameter("email\_fld");

String text=request.getParameter("feedback\_comments");

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy",

"root", "root");

st = con.createStatement();

int i = st.executeUpdate("insert into feedbacks(EmailId,FeedbackText,ReviewedFeedback) " + "values ('"+email+"','"+ text +"','"+0+"')");

if(i>0)

{

try

{

new MailSend().SendMail(email,"Gratitude!","Hi,Thank you for providing feedback to us. we would soon reply u");

}

catch (Exception e){}

request.getSession().setAttribute("feed\_error\_msg", "success");

response.sendRedirect("feedback.jsp");

}

else

{

request.getSession().setAttribute("feed\_error\_msg", "unable to take ur feedback, plz re-enter!");

response.sendRedirect("feedback.jsp");

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (st != null) {

try {

st.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

}

**Feedback Response:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

import javax.servlet.ServletException;

public class FeedbackResponse extends HttpServlet

{

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

Statement st=null;

Connection con=null;

String feed\_Id= request.getParameter("feedback\_id");

int feedback\_id=Integer.parseInt(feed\_Id);

String text=request.getParameter("feedback\_response\_text");

String email=request.getParameter("user\_email");

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy",

"root", "root");

st = con.createStatement();

int i = st.executeUpdate("update feedbacks set feedbackresponse='"+text+"' where feedbackId="+feedback\_id);

int j = st.executeUpdate("update feedbacks set reviewedfeedback='"+1+"' where feedbackId="+feedback\_id);

if(i>0 && j>0)

{

try

{

new MailSend().SendMail(email,"Reply to your Feedback","Hi,admin says user :"+text);

}

catch (Exception e)

{

}

response.sendRedirect("managefeedback.jsp");

}

else

{

request.getSession().setAttribute("feed\_res\_error\_msg", "unable to send ur feedback response, plz try again!");

response.sendRedirect("managefeedback.jsp");

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (st != null) {

try {

st.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

}

**Verify User:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

public class VerifyUser extends HttpServlet

{

Statement st=null;

Connection con=null;

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

try

{

String VerifiedUser = request.getParameter("verifyregister");

HttpSession session = request.getSession(true);

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy",

"root", "root");

st = con.createStatement();

int i=st.executeUpdate("update members set EmailVerified=1 where Verificationcode='"+VerifiedUser+"'");

if(i>0)

{

try{

response.sendRedirect("index.jsp");

return;

}

catch(Exception e){}

}

else

{

try{

request.getSession().setAttribute("ver\_error\_msg", "Verified Code Do not Match.Plz Try Again!");

response.sendRedirect("welcome.jsp");

return;

}

catch(Exception e1){}

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (st != null) {

try {

st.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

}

**Review Auctions:** Allows admins to review posted auctions:

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class ReviewAuctions extends HttpServlet

{

Connection con=null;

PreparedStatement ps=null;

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

String author\_email=request.getParameter("author\_email");

String auction\_uuid=request.getParameter("auction\_uuid");

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy","root", "root");

String query=null;

int i=0;

if (request.getParameter("reviewed\_auction") != null)

{

query="update auction set Is\_Auction\_Authenticated=1 where auction\_uuid=?";

ps= con.prepareStatement(query);

ps.setString(1,auction\_uuid);

i = ps.executeUpdate();

if(i>0)

{

try{

new MailSend().SendMail(author\_email,"Congrats!","Your Product has been Reviewed now you will receive Bids very soon!");

}

catch(Exception e){}

try{

ServletContext context= getServletContext();

request.setAttribute("review\_msg", "Successful review!");

RequestDispatcher rd= context.getRequestDispatcher("/manageauction.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

}

if(request.getParameter("delete\_auction") != null)

{

query="delete from auction where auction\_uuid=?";

ps= con.prepareStatement(query);

ps.setString(1,auction\_uuid);

i = ps.executeUpdate();

if(i>0)

{

try{

new MailSend().SendMail(author\_email,"Regret!","we are very sorry that your product has been rejected for Auction.If you have any queries please feel free to contact our developers");

}

catch(Exception e){}

try{

ServletContext context= getServletContext();

request.setAttribute("del\_msg", "Successfully deleted!");

RequestDispatcher rd= context.getRequestDispatcher("/manageauction.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

}

}

catch (SQLException e)

{

e.printStackTrace();

try{

ServletContext context= getServletContext();

request.setAttribute("manageauctions\_error\_msg", "UnSuccessful operation! Try again.");

RequestDispatcher rd= context.getRequestDispatcher("/manageauction.jsp");

rd.forward(request, response);

}

catch(Exception e1){}

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

finally

{

if (ps != null) {

try {

ps.close();

} catch (SQLException e) { /\* ignored \*/}

}

if (con != null) {

try {

con.close();

} catch (SQLException e) { /\* ignored \*/}

}

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

}

**Forget Password:**

import java.sql.\*;

import java.util.\*;

import java.io.\*;

import javax.servlet.http.\*;

import javax.servlet.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class ForgetPass extends HttpServlet

{

public void doGet(HttpServletRequest request,HttpServletResponse response)throws IOException

{

String emailid = request.getParameter("email\_id");

final String frompassword="everythingisplanned";

final String from="bideasy1@gmail.com";

Connection con=null;

ResultSet rs=null;

Statement st=null;

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bideasy",

"root", "root");

st = con.createStatement();

HttpSession session = request.getSession(true);

rs=st.executeQuery("select \* from members where Email='"+emailid+"'");

if(rs.next())

{

String forgoten\_password=rs.getString("Password");

String subject = "Forget Password NO Worries! ";

String messageText ="Your Password is :"+forgoten\_password;

try

{

new MailSend().SendMail(emailid,subject,messageText);

}

catch (Exception e)

{

}

try{

response.sendRedirect("login.jsp");

return;

}

catch(Exception e){}

}

else

{

try{

request.getSession().setAttribute("forget\_error\_msg", "InValidEmail Re-Enter");

response.sendRedirect("forgotpassword.jsp");

return;

}

catch(Exception e){}

}

}

catch (SQLException e)

{

System.out.println("Sql exception");

e.printStackTrace();

}

catch (ClassNotFoundException e)

{

System.out.println("ClassNotFound exception");

e.printStackTrace();

}

}

public void doPost(HttpServletRequest request,HttpServletResponse response)throws IOException

{

doGet(request,response);

}

}

**Mail Send:**

import java.util.\*;

import javax.mail.\*;

import javax.mail.internet.\*;

import javax.activation.\*;

import javax.servlet.http.\*;

import javax.net.ssl.\*;

public class MailSend

{

final static String host = "smtp.gmail.com";

final static String from="bideasy1@gmail.com"; // write your email address means from email

final static String password="everythingisplanned";

static boolean sessionDebug = true;

public void SendMail(String to,String subject,String messageText)throws Exception

{

Properties props = System.getProperties();

props.put("mail.smtp.starttls.enable","true");

props.setProperty("mail.transport.protocol","smtp");

props.setProperty("mail.host",host);

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "587"); //port is 587 for TLS if you use SSL then port is 465

props.put("mail.debug", "true");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.fallback", "false");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

Session mailSession = Session.getInstance(props, new javax.mail.Authenticator() {protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(from, password);

}

});

mailSession.setDebug(sessionDebug);

Message msg = new MimeMessage(mailSession);

msg.setFrom(new InternetAddress(from));

InternetAddress[] address = {new InternetAddress(to)};

msg.setRecipients(Message.RecipientType.TO, address);

msg.setSubject(subject);

msg.setSentDate(new Date());

msg.setText(messageText);

Transport transport = mailSession.getTransport("smtp");

transport.connect(host, from, password);

transport.sendMessage(msg, msg.getAllRecipients());

transport.send(msg);

transport.close();

}

public void SendBulkMail(String to[],String subject,String messageText)throws Exception

{

Properties props = System.getProperties();

props.put("mail.smtp.starttls.enable","true");

props.setProperty("mail.transport.protocol","smtp");

props.setProperty("mail.host",host);

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "587"); //port is 587 for TLS if you use SSL then port is 465

props.put("mail.debug", "true");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.fallback", "false");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

Session mailSession = Session.getInstance(props, new javax.mail.Authenticator() {protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(from, password);

}

});

mailSession.setDebug(sessionDebug);

Message msg = new MimeMessage(mailSession);

msg.setFrom(new InternetAddress(from));

InternetAddress[] address = new InternetAddress[to.length];

for( int i=0; i < to.length; i++ )

{ // changed from a while loop

address[i] = new InternetAddress(to[i]);

}

for( int j=0; j < address.length; j++)

{ // changed from a while loop

msg.addRecipient(Message.RecipientType.TO, address[j]);

}

msg.setSubject(subject);

msg.setSentDate(new Date());

msg.setText(messageText);

Transport transport = mailSession.getTransport("smtp");

transport.connect(host, from, password);

transport.sendMessage(msg, msg.getAllRecipients());

transport.send(msg);

transport.close();

}

}

**CHAPTER 6**

**6. Integration and Testing**

**6.1 Integration**

In engineering, system integration is defined as the process of bringing together the component subsystems into one system and ensuring that the subsystems function together as a system. Technically, is the process of linking together different computing systems and software applications physically or functionally, to act as a coordinated whole.

**6.2 Testing**

Testing is one of the most important phases in software development activities. In software development lifecycle (SDLC), the main aim of testing process is the quality. The developed software is tested against attaining the required functionality and performance.

During the testing process the software is worked with some particular test cases and the output of the test cases are analyzed whether the software is working according to the expectations or not.

The success of the testing process is determining the errors is mostly dependent on the test case criteria, for testing any software we need to have a description of the expected behavior of the system and method of determining whether the observed behavior confirmed to the expected behavior.

**Levels of Testing**

Testing plays a vital role to reach the 100 percent perfectness in any system. It is the major quality control measure used to determine the status and usefulness in any system.

Testing of the new system has been done successfully in different levels. There are so many activities that help to find the errors and other qualities to reach safe state in the system implementation. The number of records increased the software and hardware was found to be functioning satisfactorily.

Since the errors in the software can be injured at any stage. So, testing has to be carried out at different levels during every stage of development. The basic levels of testing are Unit, System and Acceptance testing.

The Unit testing is carried out on coding. Here different modules are tested against the specifications produced during the design for the modules. In case of integration testing different tested modules are combined into sub systems and tested in case of the System testing the full software is tested and in the Acceptance testing the system is tested with user requirement document prepared during SRS.

There are *two* basic approaches for testing. They are:

**Functional Testing:**

In functional testing test cases are decided solely on the basis of requirements of the program or module and the internals of the program or modules are not considered for selection of test cases. This is also called Black box Testing.

**Structural Testing:**

In Structural Testing test cases are generated on actual code of the program or module to be tested. This is called White Box testing.

Structural testing encompasses three critical phases of software development and testing; yet, one or more of these phases is often deliberately bypassed, overlooked or performed in a less rigorous manner because the technical advantages are not fully considered or, more often, the cost and schedule benefits are not appreciated. While structural testing is required by the FDA for medical devices of moderate and major level of concern. Also, it should be noted that there is no fundamental difference between structural testing of software used in a medical device and that used in a manufacturing process or a manufacturer’s process or a manufacturer’s quality system (or, for that matter, in any other software)

**Test Plan**

Test plan is a general document for entire project, which defines the scope, approach to be taken and the personal responsible for different activities of testing. The inputs for forming plans are:

* Project Plan
* Requirements document
* System design

**Test Case Specifications**

Although there is one test plan for the entire project test cases have to be specified separately for each test case. Test case specification gives for each item to be testing. All test cases and outputs expected for those test cases.

**Test Case Execution and Analysis**

The steps to be performed for executing the test cases are specified in separate document called test procedure specification. The document must specify any specification requirements that exist for setting the test environment and describes the methods and formats for reporting the results of testing.

**Compilation Test:**

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

**Execution Test:**

This program was successfully loaded and executed. Because of good programming there were no execution errors. The complete performance of the project “souvenir Application” was good.

**Output Test:**

The successful output screens are placed in the output screens section above with brief explanation about each screen.

**6.2.1 Test Cases**

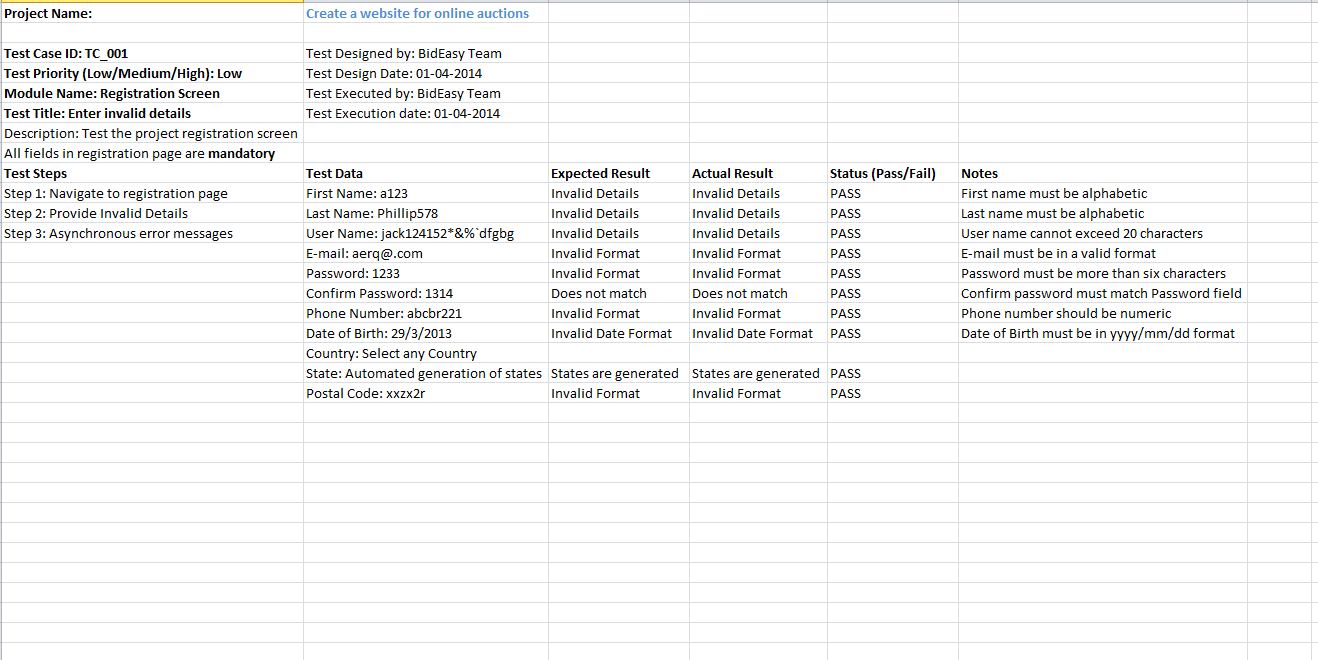


Figure 6.1: Test Case for Registration (Invalid Cases)

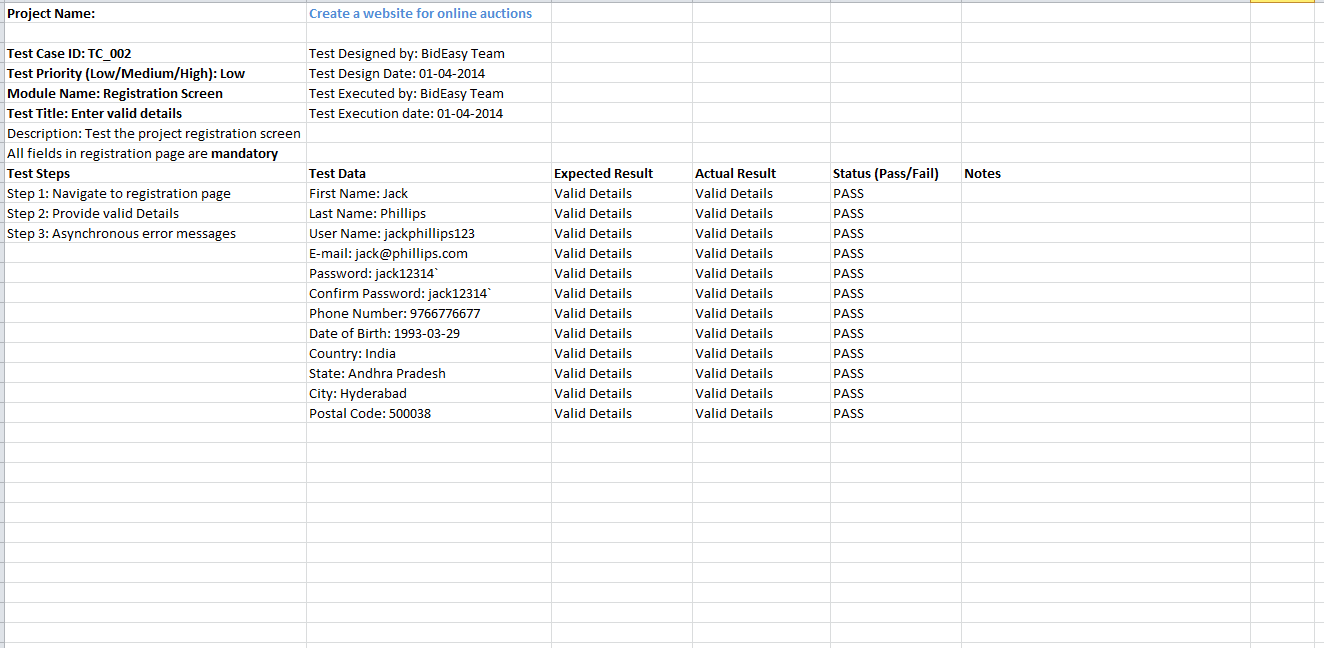


Figure 6.2: Test Case for Registration (Valid Cases)

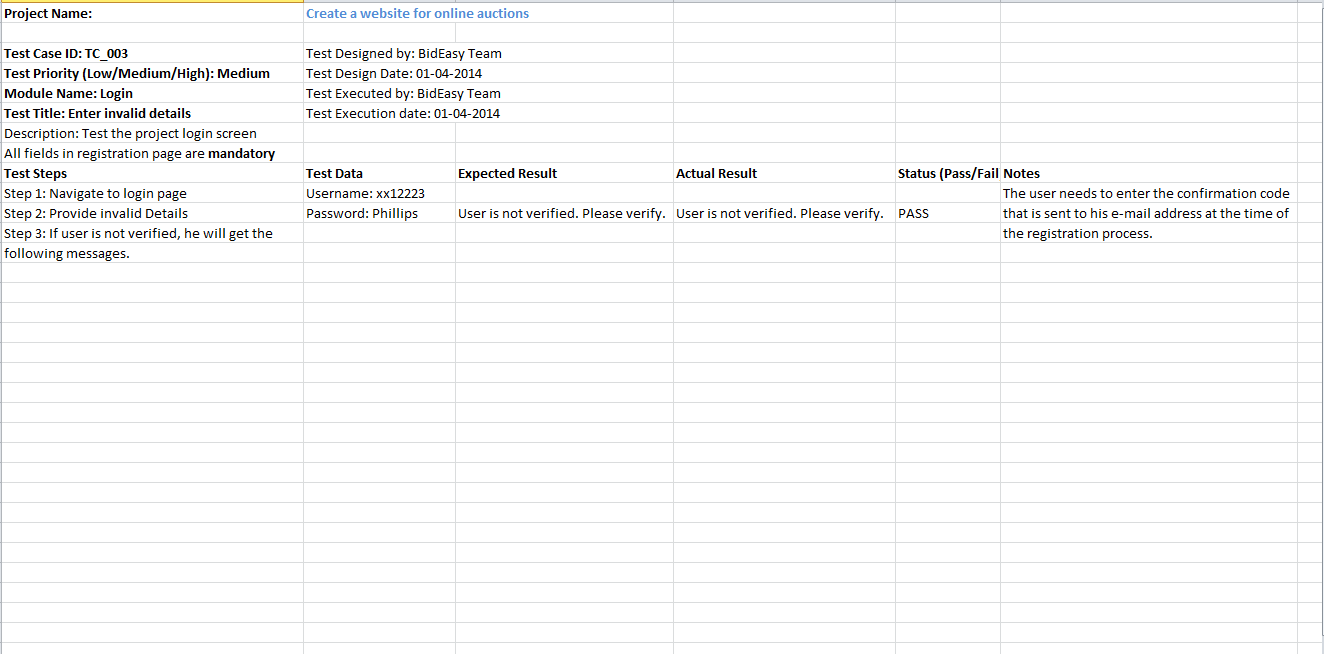


Figure 6.3: Test Case for Login (Invalid Details)

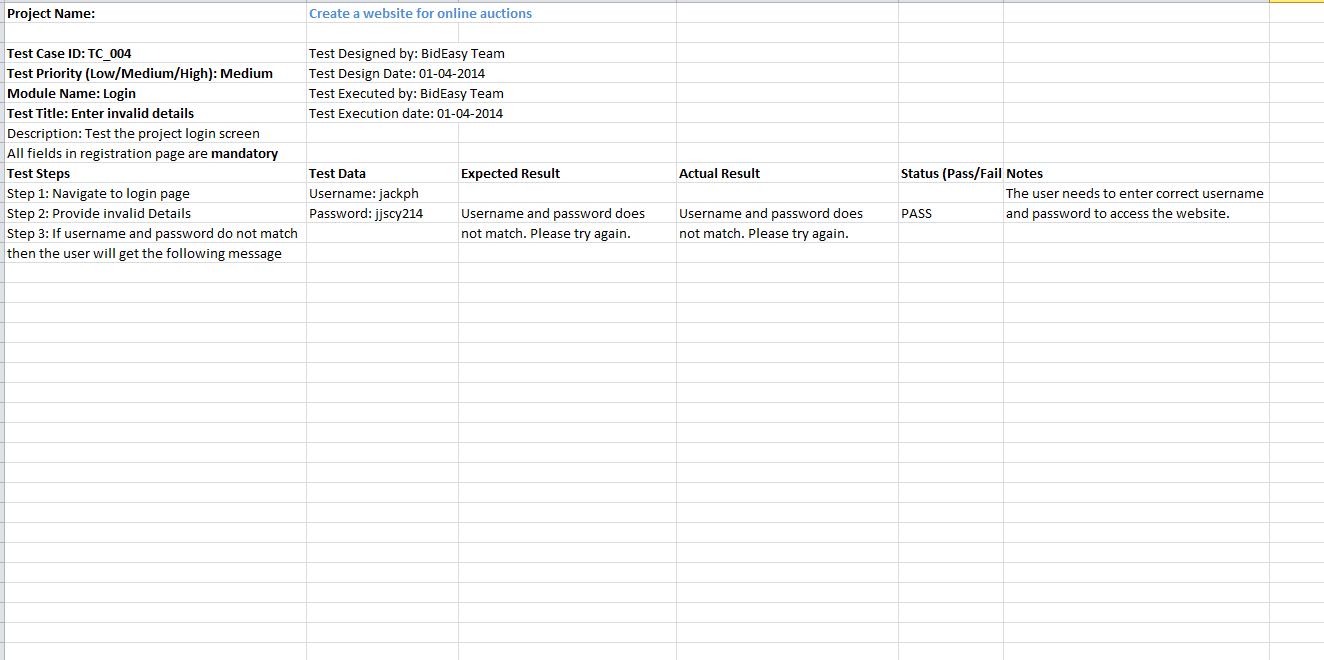


Figure 6.4: Test Case for Login (Invalid Details)



Figure 6.5: Test Case for Post Auction (Invalid Details)

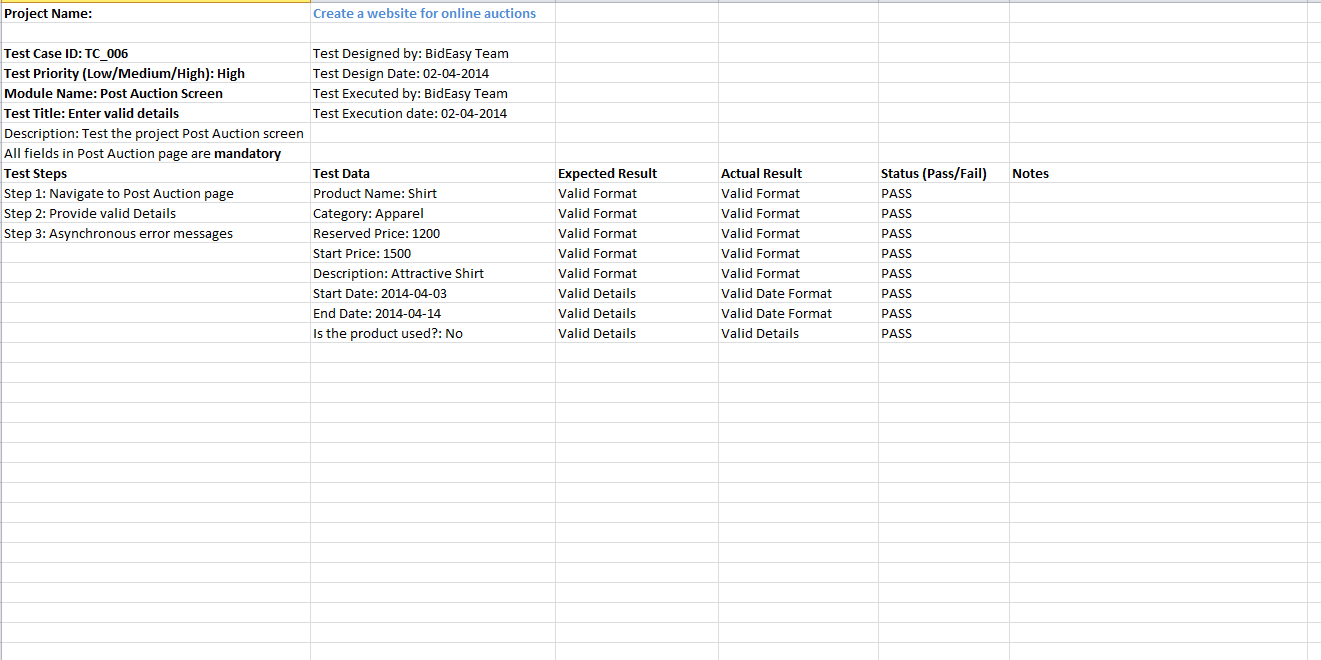


Figure 6.6: Test Case for Post Auction (Valid Details)

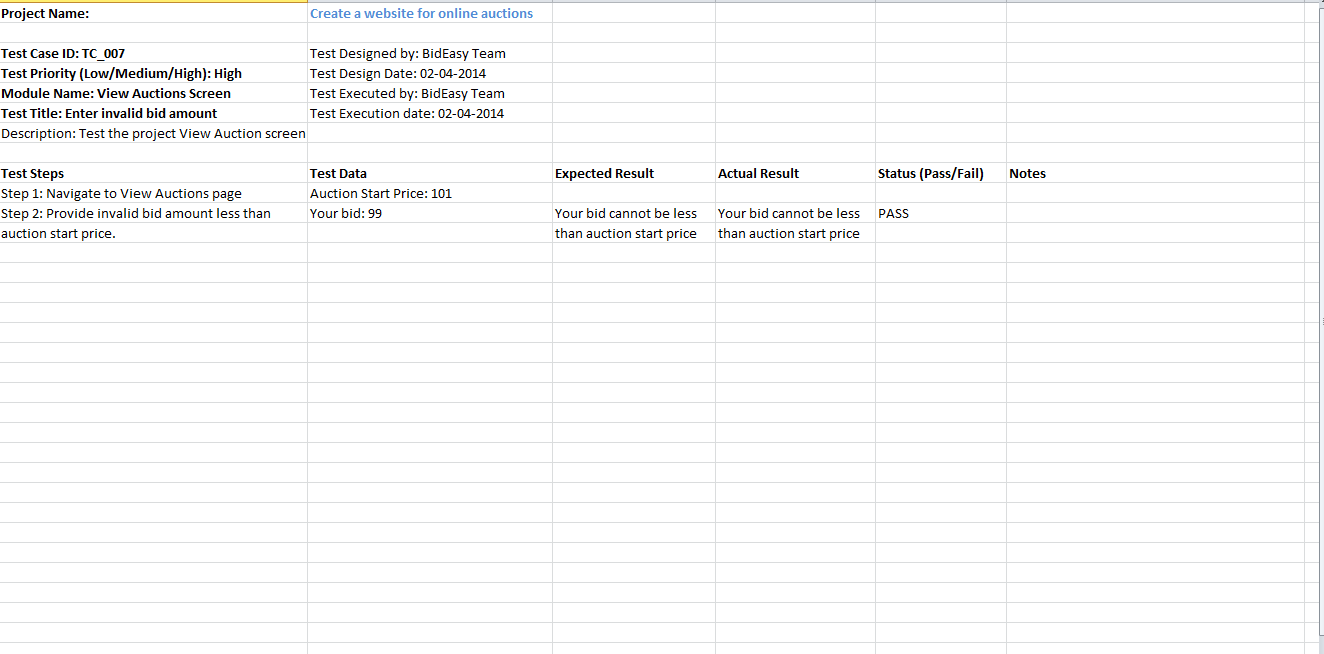


Figure 6.7: Test Case for Bid

**CHAPTER 7**

**7.1 Output Screens**

**7.1.1 Home Page**

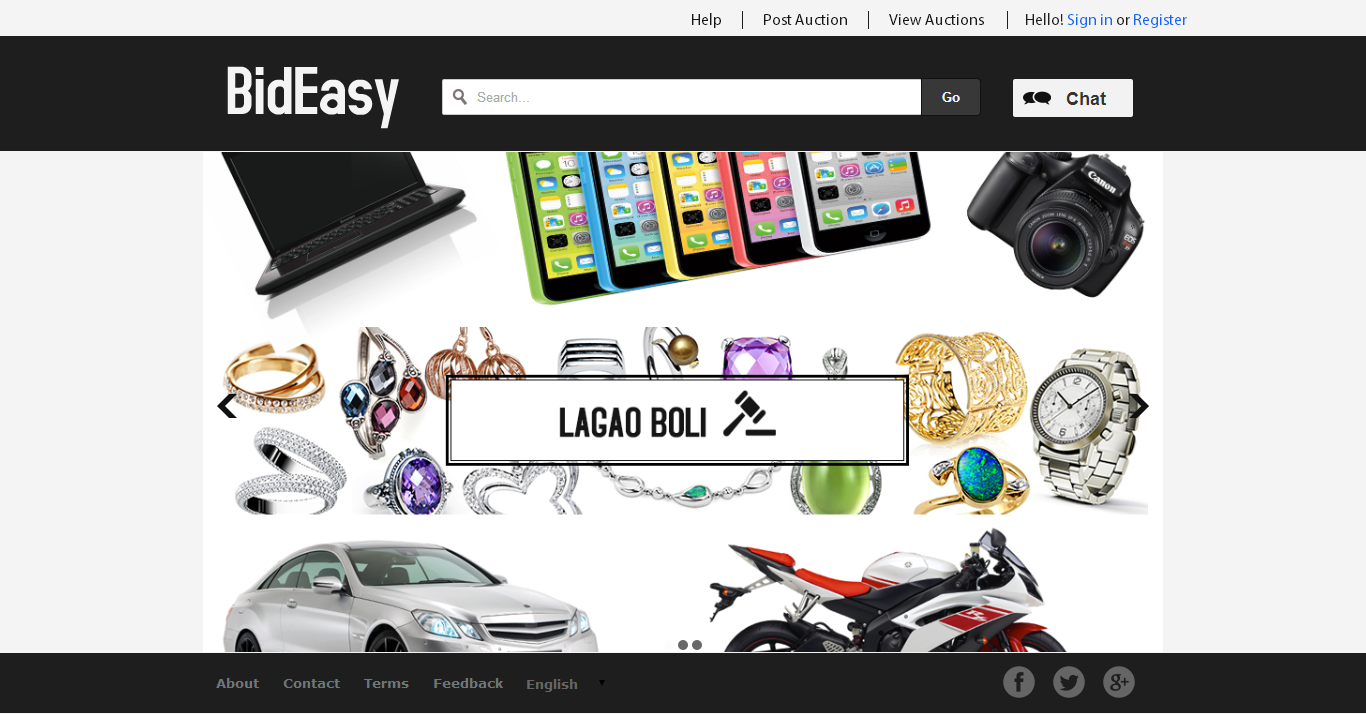


Figure 7.1: Home Page

**7.1.2 Registration Page**

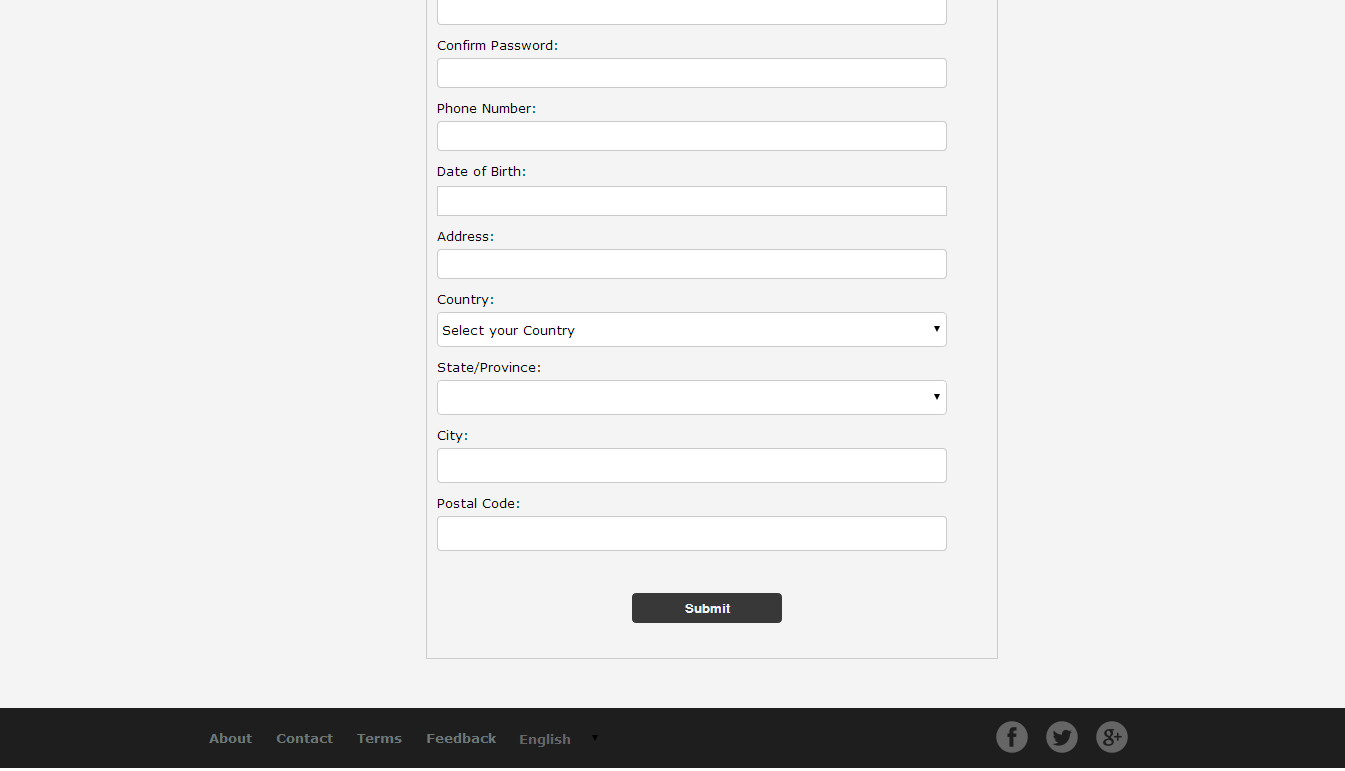
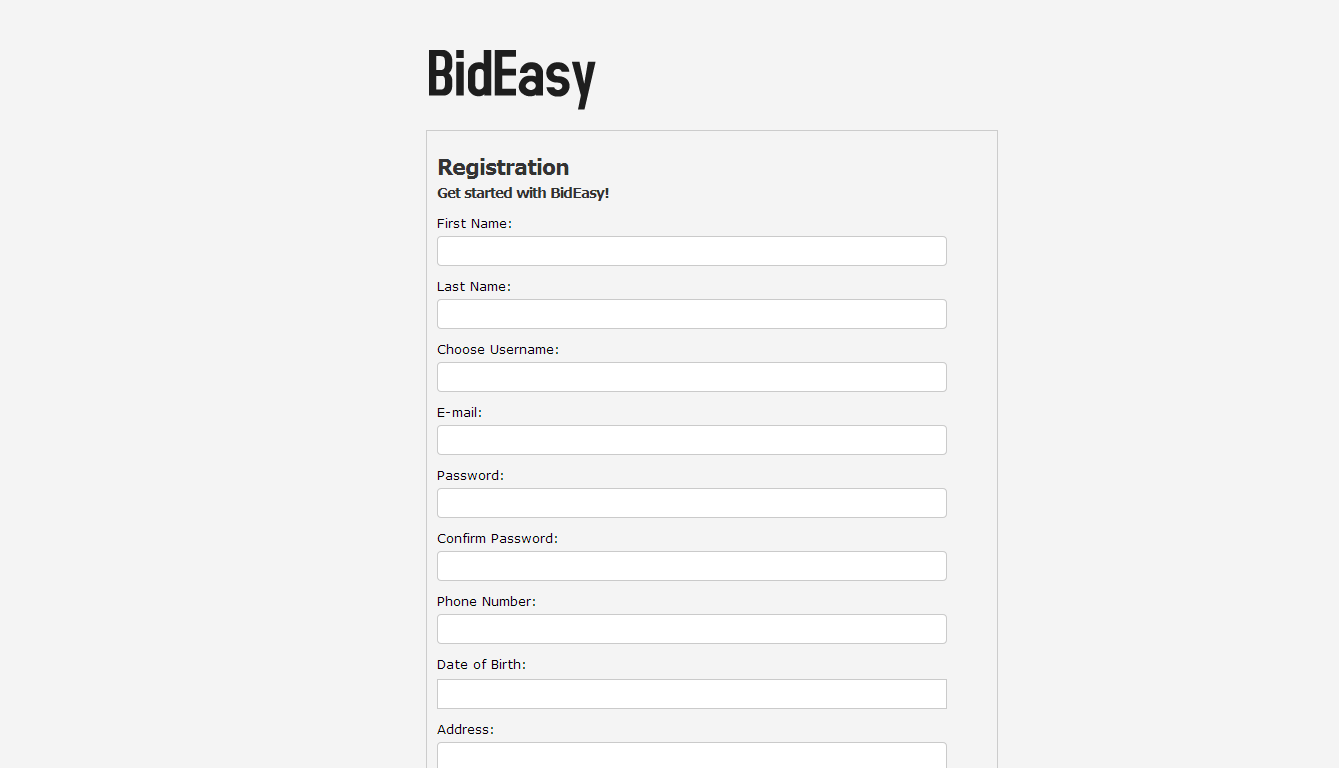


Figure 7.2: Registration Page

**7.1.3 Sign in Page**

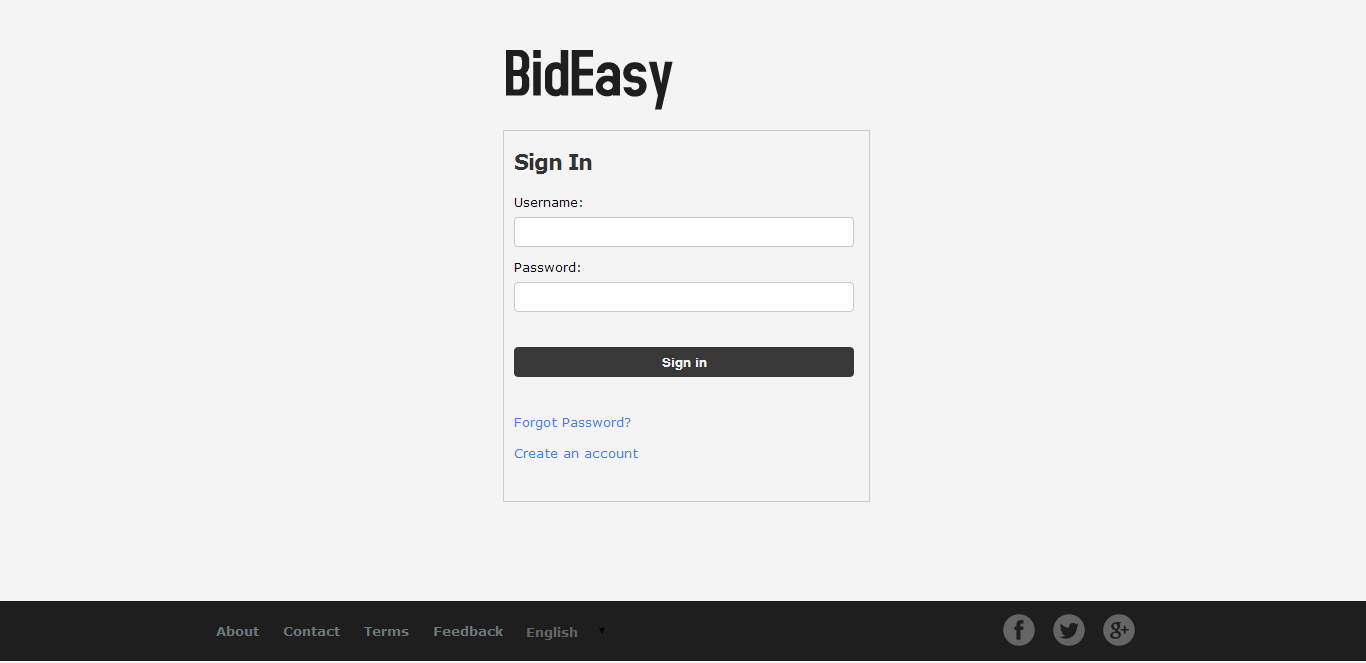
****

Figure 7.3: Sign in Page

**7.1.4 Post Auction Page**

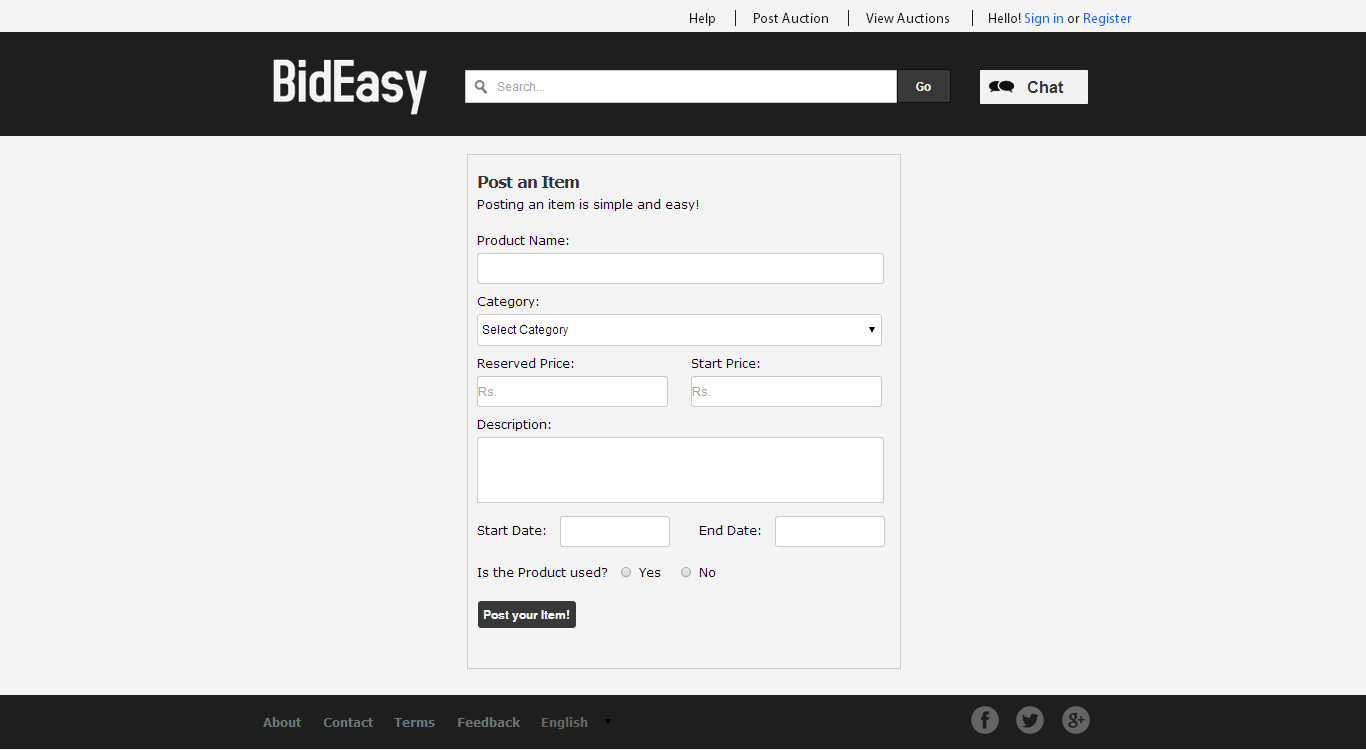


Figure 7.4: Post Auction Page

**7.1.5 View Auctions Page**

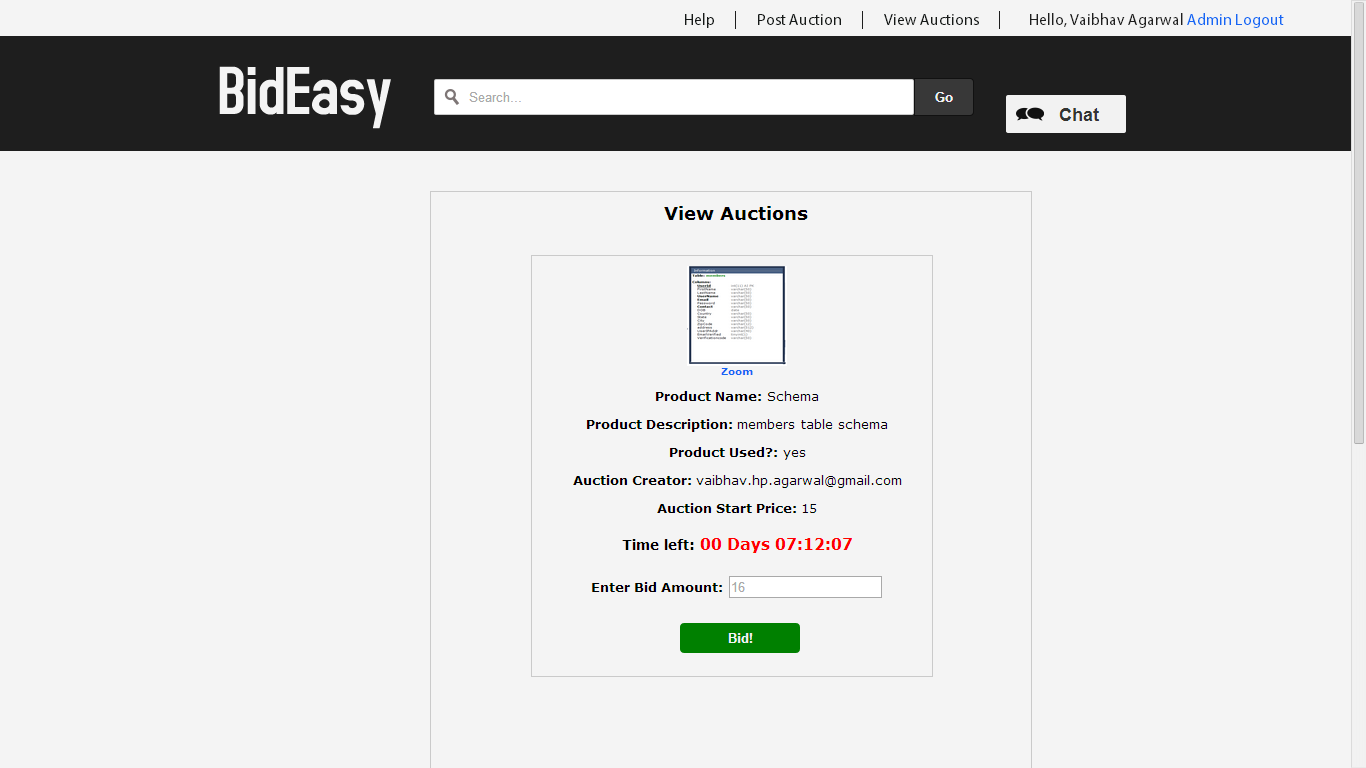


Figure 7.5: View Auctions Page

**7.1.6 Manage Auctions**

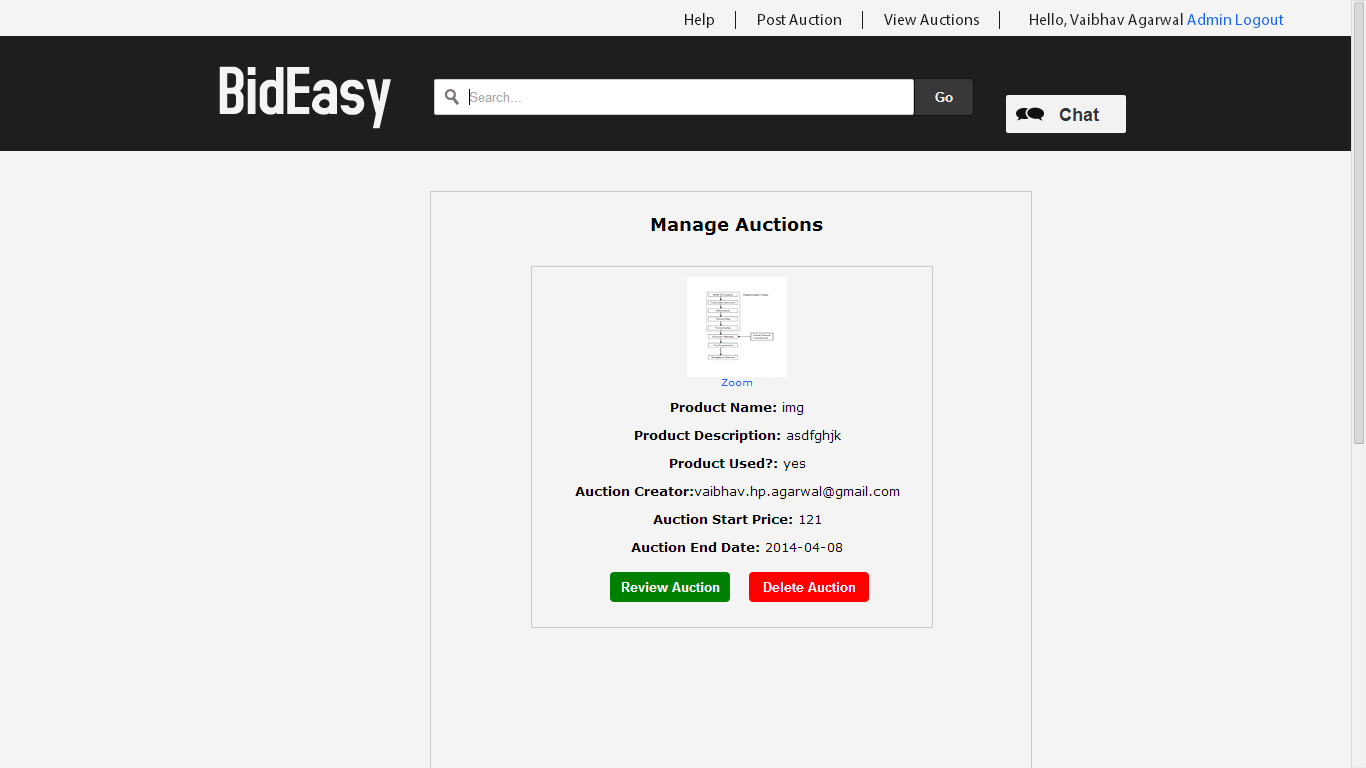


Figure 7.6: Manage Auctions Page

**7.1.7 Manage Feedbacks**

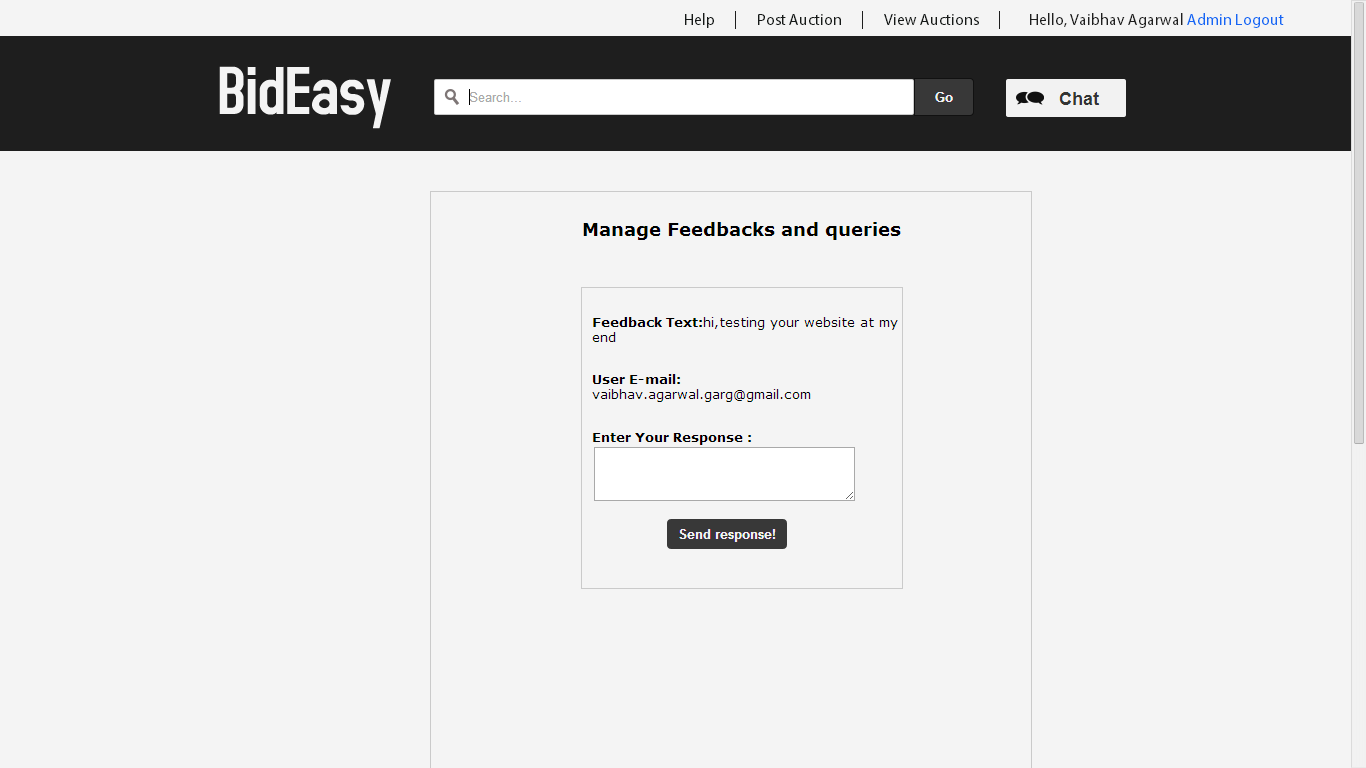


Figure 7.7: Manage Feedbacks Page

**7.1.8 About Page**

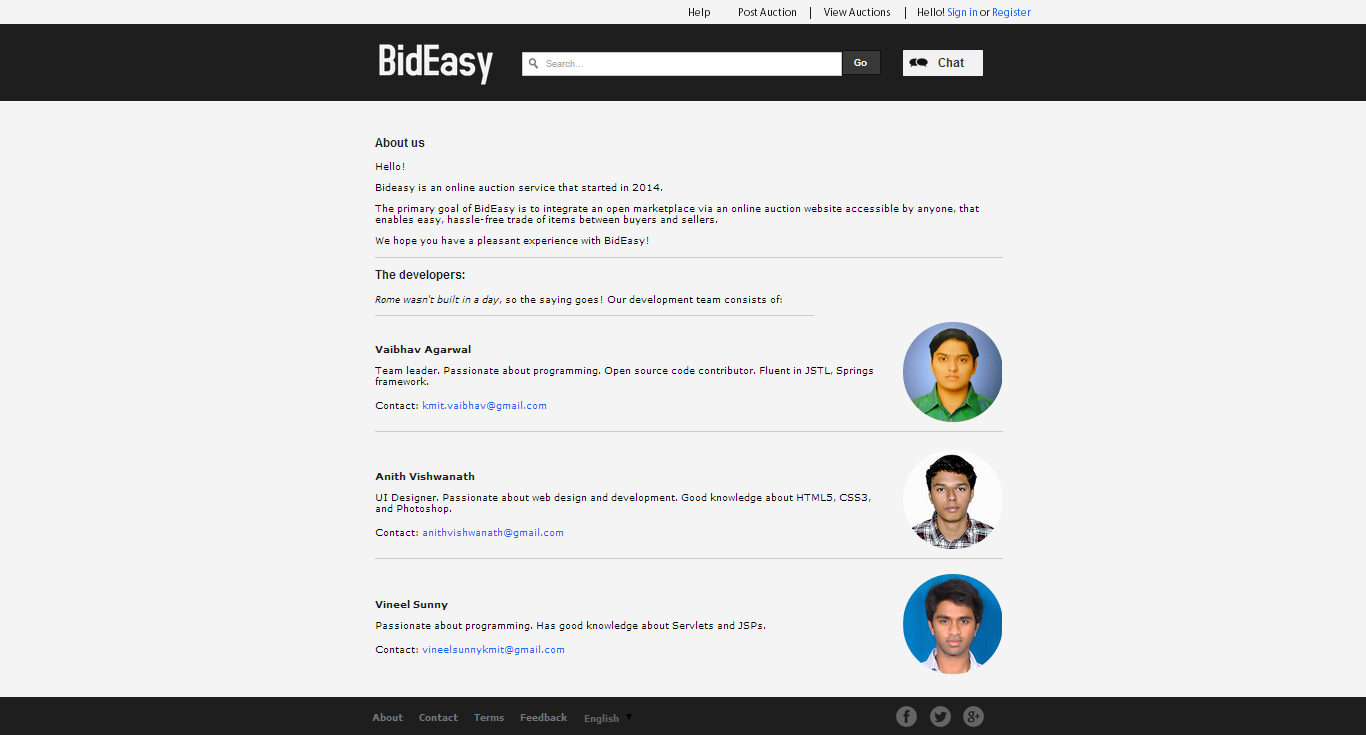
****

Figure 7.8 About Page

**7.1.9 Contact Page**

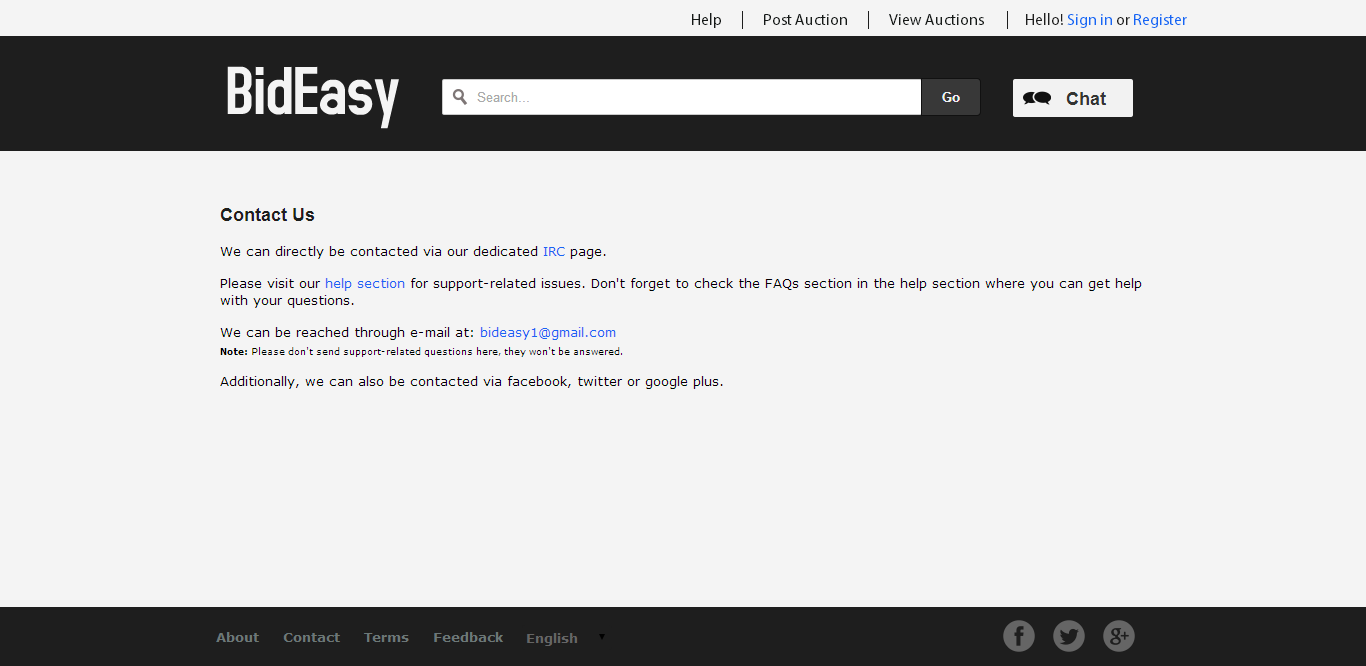
****

Figure 7.9: Contact Page

**7.1.10 Feedback Page**

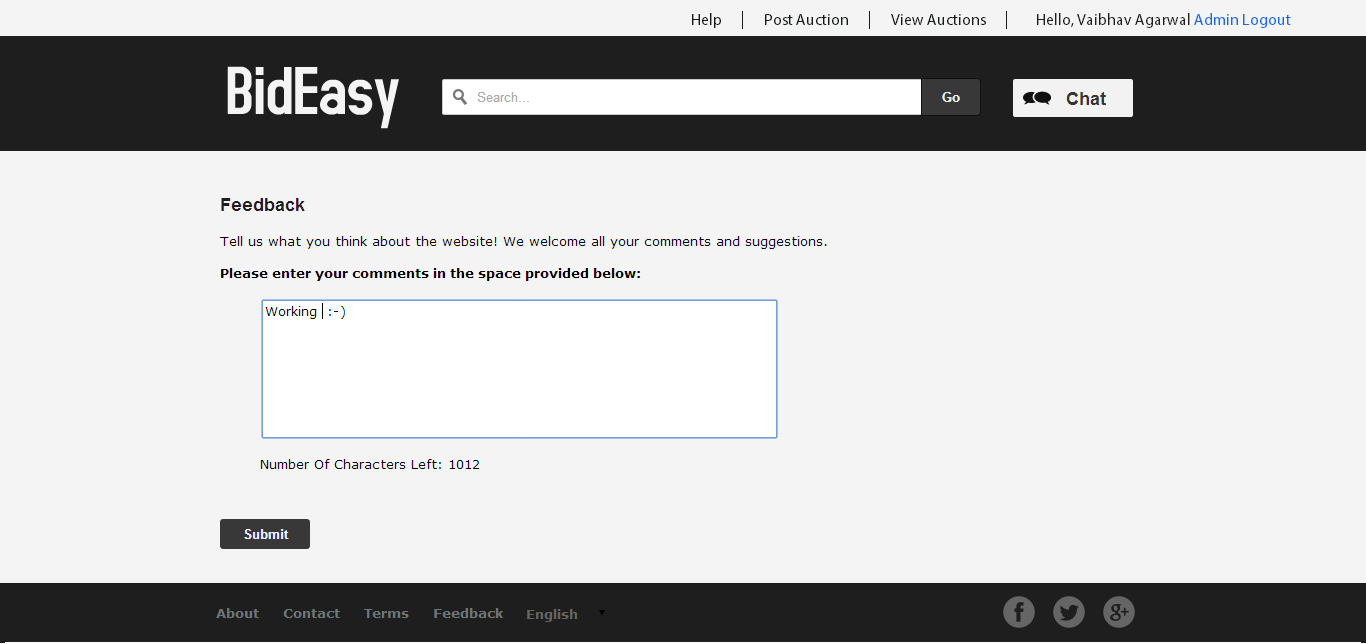


Figure 7.10: Feedback Page

**7.1.11 Chat Page**

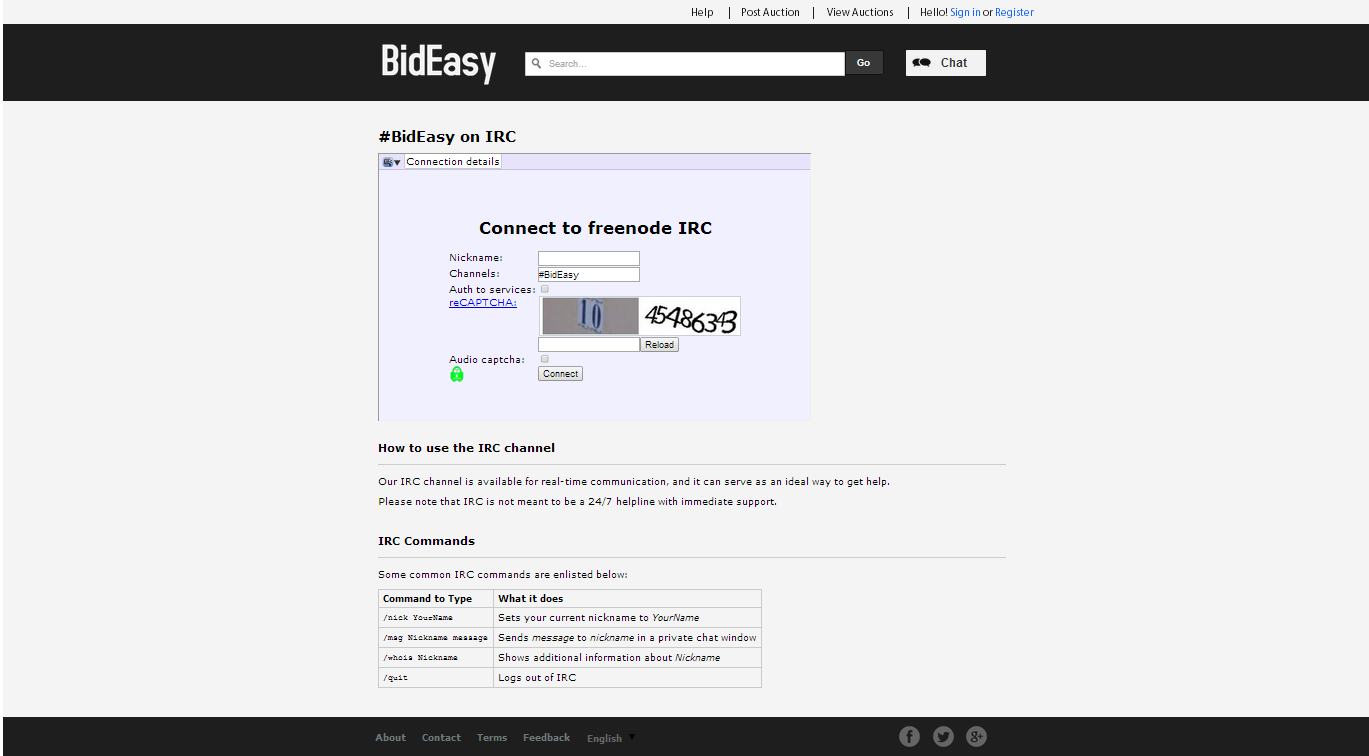
****

Figure 7.11: Chat Page

**7.1.12 Search Page**

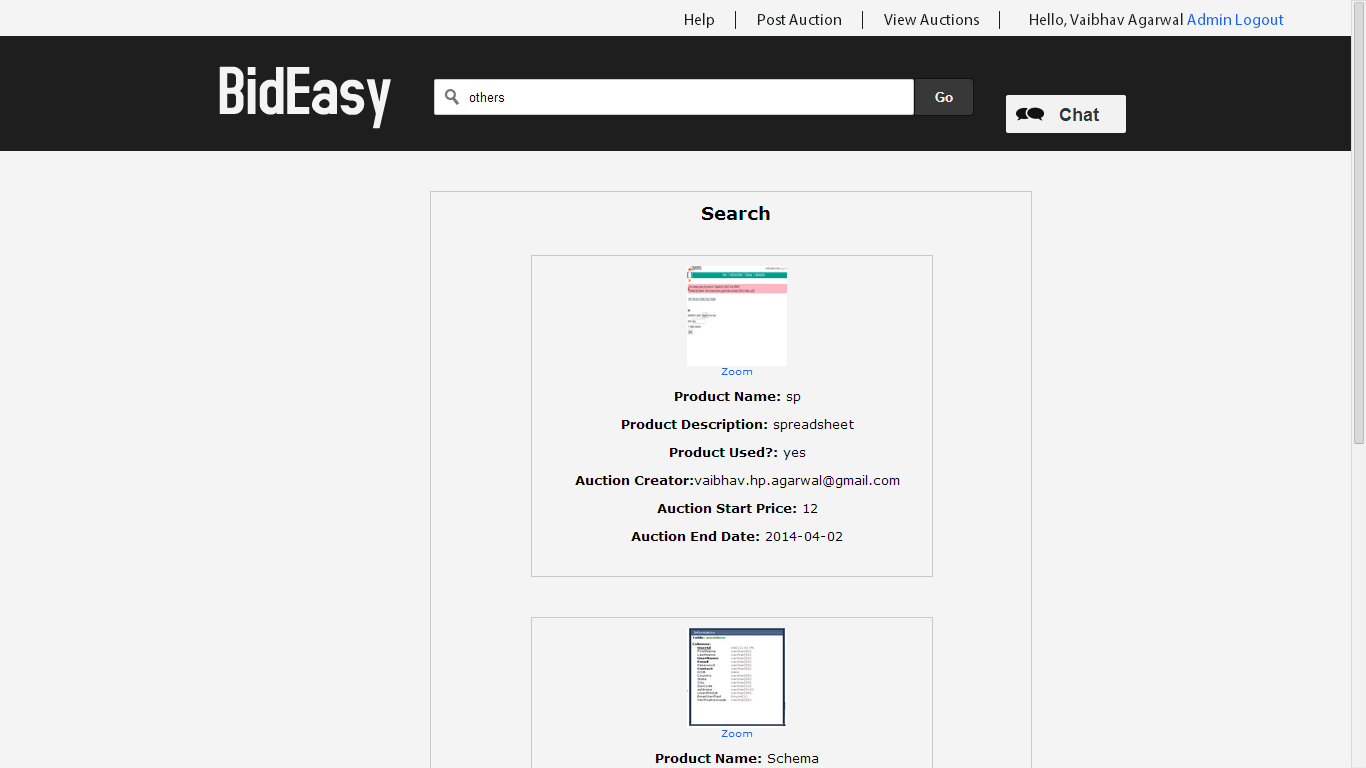
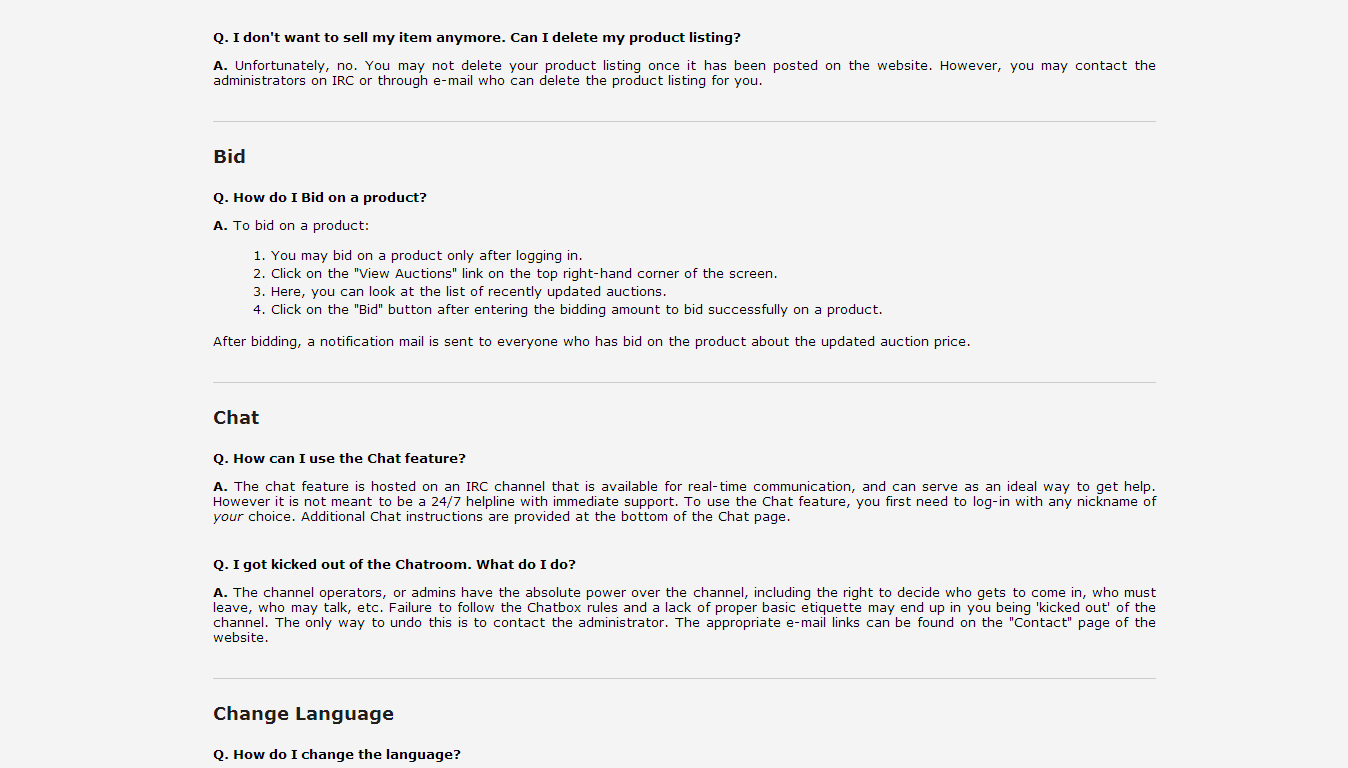
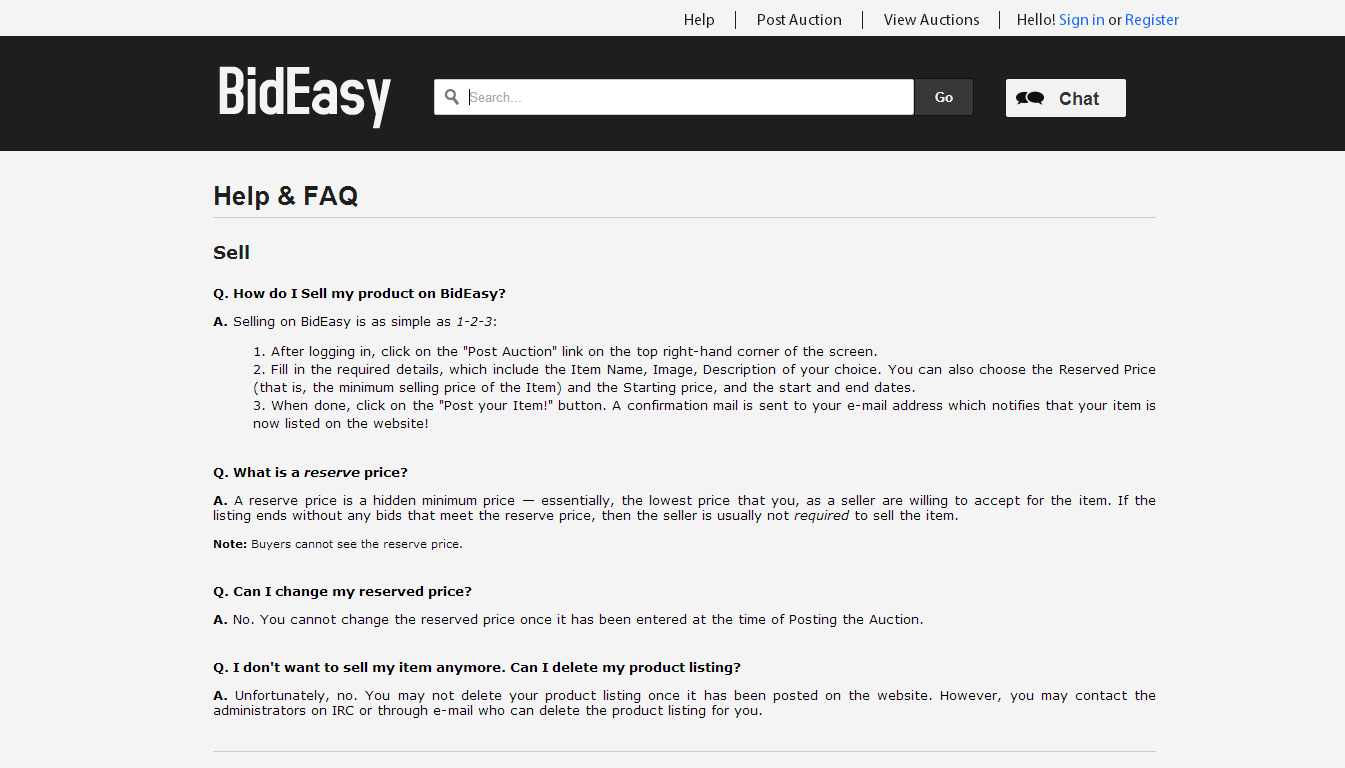
****

Figure 7.12: Search Page

**7.1.13 Help Page**

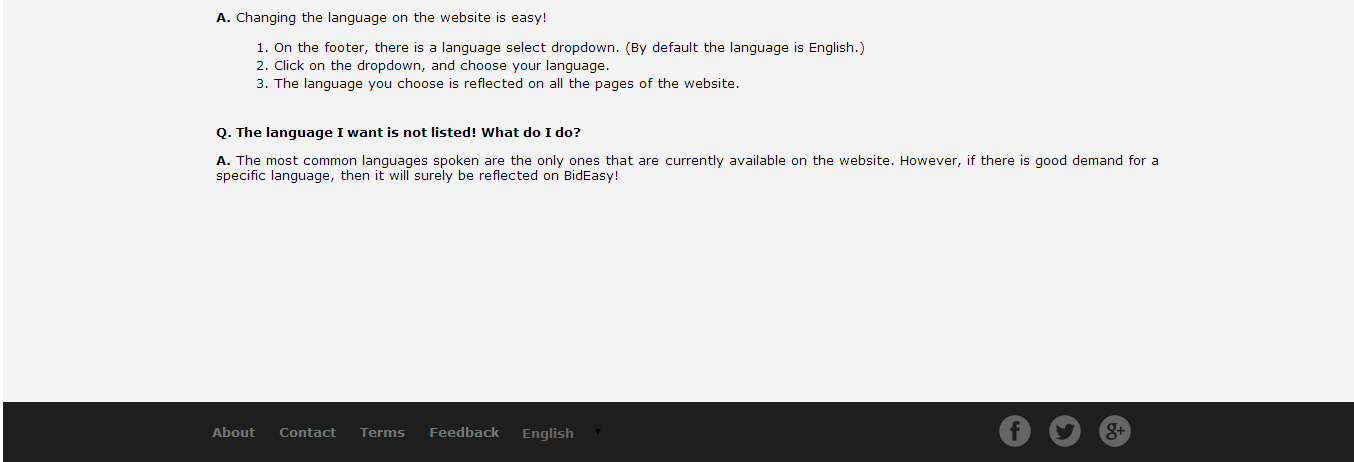
****

Figure 7.13: Help Page

**7.1.14 Internationalized page – Index**

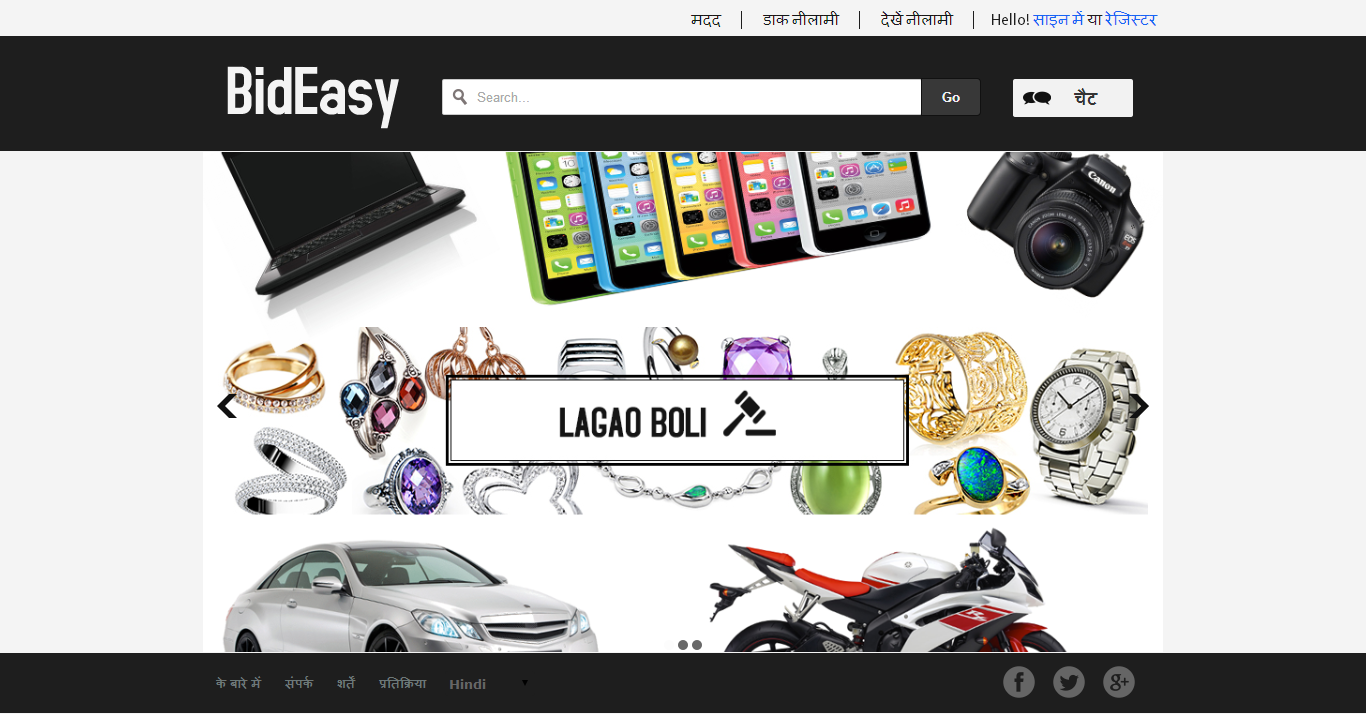
****

Figure 7.14: Internationalized Page (index page)

**7.1.15: Internationalized Page – Sign in**

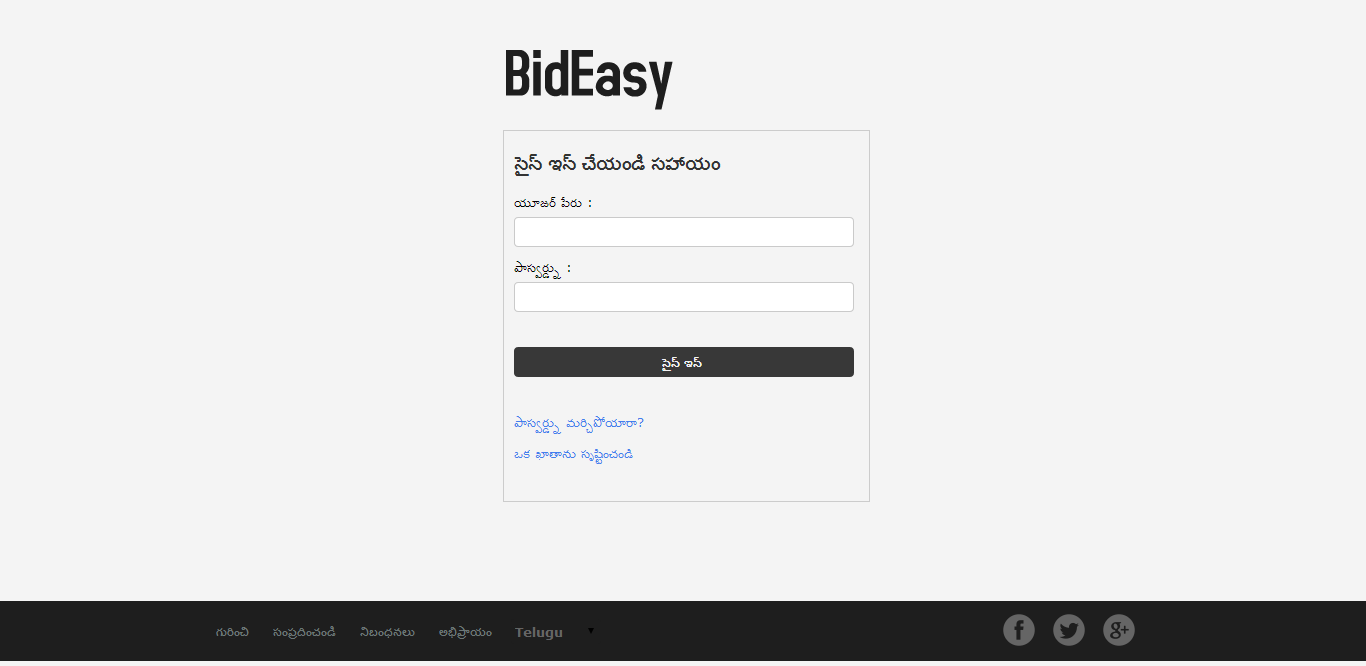
****

Figure 7.14: Internationalized Page (Sign in)

**7.2 Future Enhancements**

The online auction portal works very well in all of its functionality. However, some future works can be done on the existing system:

* Add more attractive graphical user interface (GUI) to some components of the website. The site is very easy to browse, also for new users, because the pages are simple and clear. Furthermore, there are no redundant links anywhere, and the site is fully internationalized.
* Add a Credit Card/PayPal payment system. It would be nice for users to make payments using their own Credit Card to exchange money with the help of the website.
* Work on an improved search system.
* Relocate the existing functionality and realize it on a reliable framework such as Struts or Springs.
* A notification tab that notifies users immediately and on the website instead of mass e-mails.

**7.3 Conclusion**

We conclude that our website has extra functionality compared to the existing online auction websites such as internationalization, chat, search and feedback, and also a simple module for posting and bidding of items.

**7.4 Bibliography**

1. Eclipse – <http://www.eclipse.org>

2. Oracle Java EE (version 6) API Specifications – <http://www.oracle.com>

3. Online Auction Terms – <http://en.wikipedia.org/wiki/Online_auction>

4. Online Auction Business Model – <http://en.wikipedia.org/wiki/Online_auction_business_model>

5. “Online Auctions,”by Bernhard Rumpe, Software and Systems Engineering, Munich University of Technology

6. Webopedia – <http://www.webopedia.com>

7. “Auctions on the Internet: A Field Study,” by Carrie Beam and Arie Segev, Fisher Center for Management and Information Technology, University of California, Nov. 1998

8. Stack Overflow – <http://stackoverflow.com>

9. DB Overflow – <http://dboverflow.com>

10. JSTL and JSPs – <http://tutorialspoint.com>

**7.5 Books**

1. Head first JSPs and Servlets

2. Web Programming by Robert Sebesta

3. Web Technologies by SciTech Publications

4. Java Standard Edition 6th Edition by Herbert Schildt

5. Java Extended Edition