**TENDER MANAGEMENT SYSTEM**

*A report Submitted*

*for completion of the degree of*

Bachelor of Engineering

By

**Shashi Raj (Roll No. :10300216014)**

**Deep Shikha (Roll No. :10300216047)**

**Ayush Vikas (Roll No. :10300216049)**

**…………………………………..**

***Under the supervision of***

**Mr. Ratan Kumar**



**HALDIA INSTITUTE OF TECHNOLOGY**

DEPARTMENT OF INFORMATION TECHNOLOGY

**HALDIA, PURBA MEDINIPUR, WEST BENGAL, INDIA**

**2019**

**DECLARATION**

I/we hereby declare that this project work titled**“****TENDER MANAGEMENT SYSTEM”** is our original work and this industrial training report is submitted by us to **Haldia Institute of Technology**, Haldia, in partial fulfilment of the requirements for the degree of **Bachelor of Technology** in **Information Technology**. This is a record of Industrial Training work carried out by us under the guidance of **Mr. Ratan Kumar**. We declare that this written submission represents our ideas in our own words.

We also declare that we have adhered to all principles of academic honesty and integrity and have not mis-represented or fabricated or falsified any idea/ data/ fact source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also invoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of the students Project Trainer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Shashi Raj (10300216014) (Mr. Ratan Kumar)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Deep Shikha (10300216047) Prof. (Dr.) Soumen Paul

Head of the Department,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dept. of IT

Ayush Vikas (10300216049) Haldia Institute of Technology,

Haldia, WB, INDIA

Date:

**Acknowledgement**

I/we would like to express our deepest appreciation to all those who provided us the ability to complete this report. We also take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination.

We extend our sincere and heartfelt thanks to our Industrial trainer, **Mr. Ratan Kumar** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. I extend our sincere thanks to our respected head of the department **Dr. Soumen Paul,** for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank friends for the support and encouragement they have given us during the course of our work.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Content** | **Page No.** |
|  | **CHAPTER 1. INTRODUCTION** | |
| **1.1** | Background | **06** |
| **1.2** | Objective | **08** |
| **1.3** | Purpose, Scope, and Applicability |  |
|  | **1.3.1** Purpose | **10** |
|  | **1.3.2** Scope | **11** |
|  | **1.3.3** Applicability | **11** |
|  |  |  |
|  | **CHAPTER 2. SURVEY OF TECHNOLOGIES** | |
| **2.1** | Java | **13** |
| **2.2** | J2EE |  |
|  | **2.2.1** Why J2EE? | **14** |
|  | **2.2.2** JDBC | **15** |
|  | **2.2.3** Servlets | **16** |
|  | **2.2.4** JSP | **18** |
| **2.3** | Front End Development |  |
|  | **2.3.1** HTML | **19** |
|  | **2.3.2** CSS | **19** |
|  | **2.3.3** Bootstrap | **20** |
| **2.4** | MySQL | **20** |
|  |  |  |
|  | **CHAPTER 3. REQUIREMENTS AND ANALYSIS** | |
| **3.1** | Problems | **22** |
| **3.2** | Requirements Specification | **22** |
| **3.3** | Functional Requirements | **23** |
| **3.4** | Non-Functional Requirements | **23** |
| **3.5** | Architecture | **24** |
| **3.4** | Software and Hardware Requirements | **25** |
|  |  |  |
|  | **CHAPTER 4.  SYSTEM DESIGN** | |
| **4.1** | Data Design | **26** |
| **4.2** | Physical Design |  |
|  | **4.2.1** Design Methodology | **26** |
|  | **4.2.2.** Administrator Module Design | **26** |
| **4.3** | Data Flow Diagram |  |
|  | **4.3.1** Context Level Data Flow Diagram | **28** |
|  | **4.3.2** First Level Data Flow Diagram | **29** |
|  | **4.3.3** Second Level Data Flow Diagram | **30** |
| **4.4** | Data Modeling | **31** |
| **4.5** | ER Diagram | **32** |
| **4.6** | List of Tables | **33** |
| **4.7** | Modules of the Project | **36** |
| **4.8** | Process Logic | **36** |
| **4.9** | User Interface Design | **38** |
| **4.10** | Security Issues | **38** |
|  |  |  |
|  | **CHAPTER 5. SYSTEM DEVELOPMENT** | |
| **5.1** | Implementation Approaches | **39** |
|  |  |  |
|  | **CHAPTER 6. TESTING** | |
| **6.1** | Testing | **40** |
| **6.2** | Output Screenshots | **42** |
|  |  |  |
|  | **CHAPTER 7. CONCLUSION** | **60** |
| **7.1** | Limitations of the System | **60** |
|  |  |  |
|  | **CHAPTER 8. FUTURE SCOPE OF THE PROJECT** | **61** |
|  |  |  |
|  | BIBLIOGRAPHY | **62** |
|  |  |  |
|  | SAMPLE CODE | **63** |

**CHAPTER: 1**

**INTRODUCTION**

Managing tenders and assigning the tenders to the perfect contractors and paying them a profitable stipend is one of the challenging things in the current growing world. As the number of companies are increasing, the projects and tenders are also accordingly increasing. We came up with a perfect and reliable solution for this challenging problem.

We have developed a website to maintain all the data of the tenders or projects of the company and all the projects assignment works are being done online. In this whole process no more offline paper work needed. Now the users can view all the available tenders on the different companies and in different department, just by logging in to our website. They can also apply to work for any tender by bidding an amount on a tender. If the bid is getting accepted by the company admin then the user will receive the acceptance status on their profile. And after that they may join the project as per the notification provided to them.

By doing this project we are trying to just remove all the paper work required for such process. Currently all the application process takes a lot of time for sending and also it takes a lot of time in getting the application approval status back, so we are just turning them into the digital world and making the company more profit and also saving a lot of time for both contractors and companies.

Tendering is the specific procedure. It is relatively simple but used on a large scale requires a substantial investment in IT equipment. The client organisation by means of an extranet upload the documentation on to the system and it is sent out.

Contractors download the information and prepare their tender prices which are submitted electronically and may be amended any time up to the tender deadline. So, e-tendering provides for tender documentation to be distributed to tenderers via a web-based system and allows transmission of amendments to documents and tenderers' queries during the tender period and the submission of the final bids.

The ease with which documents may be distributed and amended also presents a risk to the integrity of the system. Security features must be incorporated to ensure that:

* Tender documents cannot be accessed by unauthorised parties.
* Either party can deny sending or receiving documents.
* Alteration of tender documents is either impossible or easy to detect whilst also allowing the tenderers to insert their rates and prices.

**1.1 BACKGROUND**

In order to win in competitive selection, it is necessary for a company to pay more attention for preparation of the staff and their participation in different training projects and programs, trainings and seminars.

In order to provide assistance for companies in increasing their competitiveness in the market of work and services the following thematic seminars may be considered:

* Ways of increasing the Management Level.
* Obtaining Additional Permissions and Licenses.
* Assistance in Preparation of Tender Documents.
* Improving Staff’s Qualification.
* Implementation of Quality Assurance Programme, etc.

Both employers and contractors must be aware of advantages and disadvantages of e-tendering. The presence of an electronic witness to all transactions, the means to communicate quickly and the savings in time and paper are the main factors to persuade clients and their teams to use e-tendering.

For employers’ paper and copying costs are eliminated, document issues are in good control. Besides, e-tendering provides fast, reliable exchange of information, a transparent audit trial, clearer comparison of bids, etc. However, IT system could crash preventing distribution of documents and considerable investment is required in IT.

For Contractors and Sub-contractors, the advantages are almost the same, but IT system could crash preventing submission of tender and there is no cost saving as it will often require paper documents.

The systems are designed to improve communication between the teams working on a project, reducing potential risk and helping to ensure that the project is delivered on time. All documents, drawings, etc. may be used by all participants in the project. Users can obtain access through standard equipment.

The acceptability of Online Tendering has been on a rise in recent times. This is because more and more organizations are realizing the potential of Online Tendering in curbing irregularities in supply chain and streamlining the whole procurement process. Also, there are instances of organizations reaping huge benefits as a result of successful implementation of Online Tendering platforms. This fact has enthused many other organizations to follow suit.

An Online Tendering platform can prove to be a boon for supply chain management of any enterprise. The paper based tendering process used to be cumbersome and had a number of loopholes. It used to be governed by the whims of officials. As the Government is realizing that the red tape is impeding the whole process and going digital can bring about a huge difference, it is implementing Online Tendering platforms in more and more organizations, ministries and departments. The unnecessary time that used to get wasted in the paper-based tendering process can be saved using e-Tendering. Also, the platform assures that the whole process is secure and that the information doesn’t get into wrong hands.

Keeping all these advantages of Online Tendering platform in consideration, the government is making concerted effort to execute more and more tenders electronically. The lowering of the threshold value for mandatory Online Tendering Management brings a larger number of tenders under the purview of Tendering. This leaves even less scope for irregularities.

**1.2 OBJECTIVE:**

The main objective of this project is replacing the paper work for the tender management by the digital technology. By doing so we are also saving a lot of time of both company as well as contractor.

Along with saving time we also provide the company a lot of contractor bits in a short time period and a chance to select which thing is important for that project: i.e. either the deadline is important or the project budget or both. If the project deadline is important for the admin then it will accept the project bids from the vendors who are ready to complete the work early and also charging less for that work. And if the project budget is more important for the company then the admin may approve the bids of the contractors who bids with lowest bid amount. So, in this way our algorithm helps in making a profit for the organization and also saves a lot of time and work for both vendor and company.

Online Tendering means carrying out all traditional tendering activities in an electronic form, using the Internet. The basic objective for online tender management system is to bring security, authenticity and accountability into the tendering process. Additional benefits include savings on time, efforts and money involved in the procurement cycle for both the supplier and the contractor. Online tendering offers comparative and qualitative comparisons amongst bidders to facilitate an accurate evaluation process.

True to its customer commitment, we offer online tendering solutions that can be integrated with the existing applications without disturbing work flow, and making the process secured.

While maintaining the authenticity of signed documents, our solutions eliminate the possibility of repudiation by the sender at any stage. In addition, the sent data can be verified and authenticated for the further processing. At this site, customers not only get customized solutions, but also a guarantee for secure transactions.

In addition to compliance with CVC guidelines, the primary objective of the project was to create a favourable image of the organization by improving transparency and accountability in the public procurement process. This in turn should motivate suppliers to participate and empower officials of the organization to cut down the cost and bring internal efficiency and economy by using automation of the complex manual activities. The system should introduce a paperless environment in the organization by covering end-to-end activities in the tendering process and yet provide enough control over planning and management of different tendering activities.

**The following are the main objectives for the project:**

* Compliance with the government regulations
* Increase in efficiency and productivity
* Transparency and accountability in the process
* Expanding the accessibility of Tender Notice / advertisement
* Speed up the process
* Vendor relationship management
* Cost savings for supplier and sourcing agency
* Improved communication

**The proposed solution offers substantial benefits to the suppliers**

* **Information** on all the tenders is available at one place to the suppliers, so that they can decide, which tenders they need to participate in. Email alerts are sent to the registered suppliers whenever a tender of their interest is published on the Marketplace
* **Reduction in the transaction cost** as the requirement of unnecessary trips to the department and communication to find the required information gets eliminated
* **Reduction in stationery cost** as the supplier/contractor no longer needs to submit bulky multiple copies of his bid in the paper form. Now he submits his bid online and uploads the scanned copies of all the documents and certificates required to be submitted by him.
* **Real time availability of tender status** – After online bid submission, the supplier/contractor can know the status of his bid online. He need not visit the office of the department repeatedly for the same. E-Procurement system allows the supplier/contractor to **modify his bid** till the tender bid submission closing time specified by the department.
* **Increased reach** as the supplier has access to all the procurement requests of all the departments on the E-Procurement Marketplace. Leads to reduction in sales and promotion costs for the supplier
* **Increased comfort factor** that things are being handled in a transparent manner
* **Supplier empowerment** - With e-Procurement Solution, small and medium suppliers will get an opportunity to participate in tenders in a fair and fearless manner. They no longer have to worry about being discriminated or their bids being tampered with. They can now purchase and participate in tenders sitting in their offices

**1.3 PURPOSE, SCOPE, AND APPLICABILITY**

**1.3.1 PURPOSE**

1. The current manual system has a lot of paper work and it does not deal with better deals and management.
2. To maintain the records of tenders and vendors manually, is a Time-consuming job.
3. With the increase in database, it will become a massive job to maintain the database.
4. Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previously sale type.
5. The retrieval of records of previously sold type will be a tedious job.
6. Lack of security for the records, anyone disarrange the records of your system.
7. If someone wants to check the details of the available product for his satisfaction (like the details of the owner of the type) the previous system does not provide any necessary detail of this type.
8. Manual systems have some merit but are prone to errors and cannot deal with growing volumes of tenders to be managed by both buyers and sellers.

**1.3.2 Scope**

1. Problem of Reliability: Current system is not reliable. It seems to vary in from one month to the, next. Sometimes it gives quality good profits, but sometimes the profit and management are worst.
2. Problem of Accuracy: There are too much tender approval delay and current process cannot deal with growing volumes of tenders to be managed by both buyers and sellers.
3. Problem of timeliness: In the current system the tender approval or rejection status comes mostly late and in most of the cases it is useless because it is not on time.
4. Problem of Validity: The tender may get expire or assigned to any other contractor due to delay or misleading information. The contractor’s information is sometimes not valid.
5. Problem of Economy: The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired profits.
6. Problem of Capacity: The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work

**1.3.3 Applicability**

1. Details: The new proposed system stores and maintains all the details of the tenders, companies, vendors, contractors, etc.
2. Registers: There is no need of keeping and maintaining database manually. It remembers each and every record and we can get any report related to tenders, vendors and contractors at any time.
3. Speed: The new proposed system is very fast with 100% accuracy and saves time.
4. Manpower: The new proposed system needs less manpower. Less people can do the large work.
5. Efficiency: The new proposed systems complete the work of many persons in less time.
6. Past details: the new proposed system contains the details of every past user for future assistance.
7. Reduces redundancy: The most important benefit of this system is that it reduces the redundancy of data within the data.
8. Work load: Reduces the work load of the data store by helping in easy updates of the records and providing them with the necessary details together with database management system.

**CHAPTER: 2**

**SURVEY OF TECHNOLOGIES**

**2.1 Java**

Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.

One of the main reasons Java is so popular is its platform independence, which means that Java programs can be run on many different types of computers. A Java program runs on any computer with a Java Runtime Environment, also known as a JRE, installed. A JRE is available for almost every type of computer — PCs running Windows, Macintosh computers, Unix or Linux computers, huge mainframe computers, and even cell phones.

Java is inherently object-oriented, which means that Java programs are made up of programming elements called objects. Simply put, an object is a programming entity that represents either some real-world object or an abstract concept.

All objects have two basic characteristics:

* Objects have data, also known as state.

For example, an object that represents a book has data such as the book’s title, author, and publisher.

* Objects also have behavior, which means that they can perform certain tasks. In Java, these tasks are called methods.

For example, an object that represents a car might have methods such as start, stop, drive, or crash.

Classes are closely related to objects. A class is the program code you write to create objects. The class describes the data and methods that define the object’s state and behavior. Then, when the program executes, classes are used to create objects.

**2.2 J2EE**

J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitiered, Web-based applications.

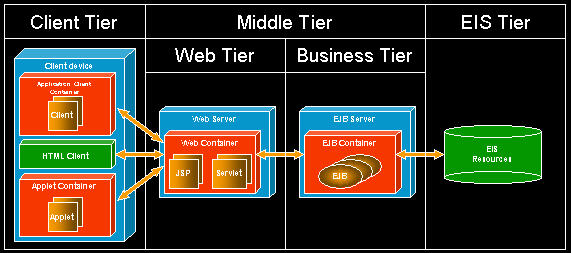


Figure 1: J2EE Architecture

**2.2.1 Why J2EE?**

J2EE Stands for Java 2 Enterprise Edition. J2EE is an environment for developing and deploying enterprise applications. J2EE specification is defined by Sun Microsystems Inc. The J2EE platform is one of the best Platform for the development and deployment of enterprise applications. The J2EE platform is consists of a set of services, application programming interfaces (APIs), and protocols, which provides the functionality necessary for developing multi-tiered, web-based applications. You can download the J2EE SDK and development tools from Oracle Technology Network for Java Developers.

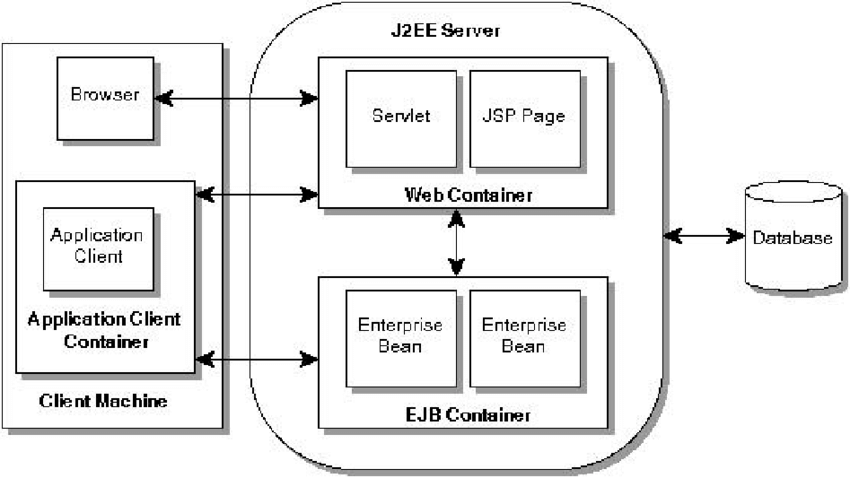


Figure 2: J2EE Connectivity and Process Flow

Core java is the basic java programming and is called as the J2SE (Java standard edition). This is basically used to develop standalone programs. Based on this sun had released the J2EE(Java enterprise edition). This is basically used to develop the enterprise applications and also web applications. J2EE consists of Servlets, JDBC, Servlets, JSP EJB.

**2.2.2 JDBC**

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

JDBC ('Java Database Connectivity') allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection.

JDBC represents statements using one of the following classes:

**Statement** – the statement is sent to the database server each and every time.

**PreparedStatement** – the statement is cached and then the execution path is pre-determined on the database server allowing it to be executed multiple times in an efficient manner.

**CallableStatement** – used for executing stored procedures on the database.

Update statements such as **INSERT**, **UPDATE** and **DELETE** return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

There is an extension to the basic JDBC API in the javax.sql.

JDBC connections are often managed via a connection pool rather than obtained directly from the driver.

**2.2.3 Servlets**

A Java servlet is a Java software component that extends the capabilities of a server. Although servlets can respond to many types of requests, they most commonly implement web containers for hosting web applications on web servers and thus qualify as a server-side servlet web API. Such web servlets are the Java counterpart to other dynamic web content technologies such as PHP and ASP.NET.

A Java servlet processes or stores a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP. Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML and more commonly, JSON. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL mapping.

The Java servlet API has, to some extent, been superseded by two standard Java technologies for web services:

The Java API for RESTful Web Services (JAX-RS 2.0) useful for AJAX, JSON and REST services, and the Java API for XML Web Services (JAX-WS) useful for SOAP Web Services.

To deploy and run a servlet, a web container must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of the web container and a servlet.

A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package javax.servlet.http defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a web application.

Servlets can be generated automatically from Java Server Pages (JSP) by the Java Server Pages compiler. The difference between servlets and JSP is that servlets typically embed HTML inside Java code, while JSPs embed Java code in HTML. While the direct usage of servlets to generate HTML (as shown in the example below) has become rare, the higher level MVC web framework in Java EE (JSF) still explicitly uses the servlet technology for the low-level request/response handling via the Faces Servlet. A somewhat older usage is to use servlets in conjunction with JSPs in a pattern called "Model 2", which is a flavor of the model–view–controller.

**2.2.4 JSP**

Java Server Pages (JSP) is a collection of technologies that helps software developers create dynamically generated web pages based on HTML, XML, SOAP, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but it uses the Java programming language.

To deploy and run Java Server Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime, therefore JSP is a Servlet; each JSP servlet is cached and re-used until the original JSP is modified.

Java Server Pages can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML. The resulting page is compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other .jar or Java program, code must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well.

The Web container creates JSP implicit objects like request, response, session, application, config, page, page Context, out and exception. JSP Engine creates these objects during translation phase.

JSP pages use several delimiters for scripting functions. The most basic is <% ... %>, which encloses a JSP scriptlet. A scriptlet is a fragment of Java code that is run when the user requests the page. Other common delimiters include <%= ... %> for expressions, where the scriptlet and delimiters are replaced with the result of evaluating the expression, and directives, denoted with <%@ ... %>.

**2.3 FRONT END DEVELOPMENT**

**2.3.1 HTML**

Hyper Text Markup Language and Extensible markup Language are the predominant markup languages for web pages. It provides a means to describe the structure of text-based information in a document and to supplement that text with interactive forms, embedded images, and other objects.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**2.3.2 CSS**

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

**2.3.3 Bootstrap**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

Bootstrap is the third-most-starred project on GitHub, with more than 135,000 stars, behind only free Code Camp (almost 305,000 stars) and marginally behind Vue.js framework. According to Alexa Rank, Bootstrap getbootstrap.com is in the top-2000 in US while vuejs.org is in top-7000 in US.

**2.4 DATABASE TECHNOLOGIES**

**2.4.1 MYSQL**

MySQL is the world’s most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after co-founder Michael Widenius daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other ‘AMP’ stacks). LAMP is an acronym for “Linux, Apache, MySQL, Perl/PHP/Python.” Free-software-open source projects that require a full-featured database management system often use MySQL.

This Project is coupled with material on how to use the various tool, sub sets available PHP and MySQL.

The need of today’s software development is competence in a GUI based front-end tool, which can connect to Relational Database engines. This gives the programmer the opportunity to develop client server based commercial applications.

**CHAPTER: 3**

**REQUIREMENTS AND ANALYSIS**

**3.1 Problems**

**Problem of Reliability:**

Current system is not reliable. It seems to vary in from one month to the, next. Sometimes it gives quality good profits, but sometimes the profit and management result is worst.

**Problem of Accuracy:**

There are too much tender approval delay and current process cannot deal with growing volumes of tenders to be managed by both Vendors and Contractors**.**

**Problem of timeliness:**

In the current system the tender approval or rejection status comes mostly late and in most of the cases it is useless because it is not on time.

**Problem of Validity:**

The tender may get expire or assigned to any other contractor due to delay or misleading information. The contractor’s information is sometimes not valid.

**Problem of Economy:**

The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired profits.

**Problem of Capacity:**

The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work

**3.2 Requirement Specification**

**Requirement Specification**

The main part of problem is to obtain a clear understanding of the needs of user and what exactly are desired from the software. It is used for specifying the requirement.

**3.3 Functional Requirements:**

* Admin or companies launch tenders with base price and last deadline for application.
* Contractors or vendors can view the available tenders.
* Vendors needs to apply for a tender by bidding a amount greater than the base price of available unassigned tenders.
* A Particular Project are being bid by different vendors with different price.
* Admin will view all the bids for a tender and can approve or reject to them.
* The approval of the tenders is leads to get the contractor who is ready to work in project with lowest stipend or bid price and/or in the nearest deadline as per requirement.
* If a project is assigned to any vendor or contractor then that project is marked as assigned and no more vendors can bid for that particular tender.
* The Tender approval or rejection status can be seen from the vendors own profile and their history section.
* The user can also register, update and modify their profile details in the sections.
* The users can easily see the assigned project notice in their profile.

**3.4 Non-Functional Requirements:**

Non-functional requirements pertain to other information needed to produce the correct system. These requirements are not functional in nature, these are the constraints within which the system must work.

**Performance**: The system can sometimes take longer time to process the approval but the project assignment should be consistent with the requirement of the company.

**Scalability**: The system should be scalable enough to store all the attributes related to the tenders and vendors.

**Usability**: User interaction has to be made comfortable and user friendly. Any one can easily search for any tender and bid for that and see the tender status.

**Reliability**: The system would assign the tenders to reliable Contractors; however human intervention would be needed in high stake application.

**Flexibility**: System can be used for launching and approving any type of tenders from a company.

**3.5 ARCHITECTURE**

Controller layer acts as an interface between View and Model. It receives requests from the View layer and processes them, including the necessary validations.

The requests are further sent to Model layer for data processing, and once they are processed, the data is sent back to the Controller and then displayed on the View.

**THE VIEW LAYER:**

This layer represents the output of the application, usually some form of UI. The presentation layer is used to display the Model data fetched by the Controller.

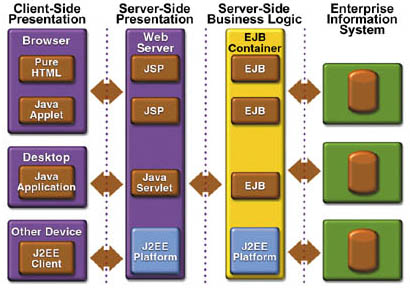
****

Figure 3: The Architecture Design

**MVC with Servlets and JSP**

To implement a web application based on MVC design pattern, we have created the *Class* and *Service* classes – which acts as our Model layer.

**3.6 SOFTWARE & HARDWARE REQUIREMENTS**

**HARDWARE:**

Processor                     :          Inter core 2 Duo or above

Memory                       :          256 GB RAM or above

Cache Memory            :          128 KB or above

Hard Disk                   :          20 GB or above [at least 3 MB free space required]

System : Pentium IV based systems

**SOFTWARE:**

Operating System       :        Windows 7 or above

Font-End Tool            :         HTML, CSS, Bootstrap

Technology : Java 1.8, JDBC, JSP, Servlet

Back-End                    :         MySQL

Server                          :         Apache Tomcat 8.0

IDE : Eclipse EE

**CHAPTER: 4**

**SYSTEM DESIGN**

**4.1 DATA DESIGN**

* **Flexible –** System can be used for management of any type of tenders bidding fields.
* **Maintainable –** The system should be scalable enough to store all the attributes related to the tenders and contractors.
* **Portable *–*** The system can take longer time to process the answer script but the grades should be consistent with the scores provided by human evaluation.
* **Easy to use –** User interaction has to be made comfortable and user friendly. For each user we have profile section, notice section and tender status section.
* **Reliable *–***The system would assign the tenders to reliable and profitable contractors; however human intervention would be needed in high stake application.

**4.2 PHYSICAL DESIGN**

**4.2.1 Design Methodology:**

This Website is Designed in a way that the user can register as a vendor and bid for the available tenders as the tender list shown and wait for the approval status to get assigned into some projects.

**4.2.2 Administrator Module:**

Administrator is the master user of this system who will responsible for uploading advertisement documents, verifying suppliers by using their information, maintaining data and confirmation of final report. The administrator module has the login and logout functionality. The admin is the person who add some project or tenders along with their details like base price, deadlines and documentation on the website. The admin has the accessibility to reject or accept the bids on any tender done by vendors.

**4.2.3 Validation Module:**

The Validation module deals with verifying whether admin or vendors details entered into the login page is valid or not. The validation module also deals with verifying whether the user is authorised vendor or not.

**4.2.4 Vendor module:**

The vendor needs to get registered and log in with a valid username and password. The vendor can apply for the tender and view the status of the tender. The vendor can bid multiple times with a valid amount on any open tender. The Vendor module also shows all the details of the bidding history and tender approval status. Vendors can also send feedback to the admin regarding their website usage and complaints.

**4.2.5 Company Module:**

Company gets username and password to login as admin and can view and manages the tenders. Company can view the details of the vendors who have been assigned to their project or tenders.

**4.3 DATA FLOW DIAGRAM**

DFDs are built utilizing four noteworthy parts:

* **External elements –**Notice Board, Tender Status, Approval
* **Data stores –**Tender, Vendor, Bidder, Admin
* **Processes –**Add-Tender, Remove-Tender, Approve-Tender, Update-Profile, Bid-Status

Process names are by and large unambiguous and pass on however much importance as could reasonably be expected without being too long. Case: check data, obtained time plan and so on.

**Data flows –** Speaks to the development of data between different parts.

The Data Flow Diagram for this tender management System is shown below:

* Context Level or Zero Level Data Flow Diagram
* Fist Level Data Flow Diagram
* Second Level Data Flow Diagram

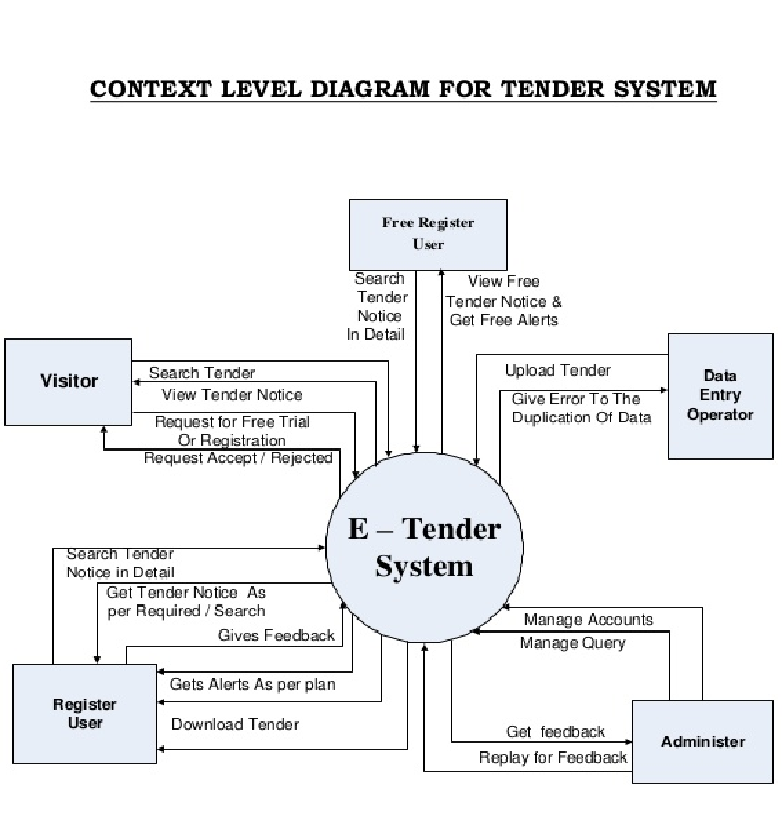
**4.3.1 CONTEXT LEVEL DATA FLOW DIAGRAM**

Figure 4: Context Level Data Flow Diagram

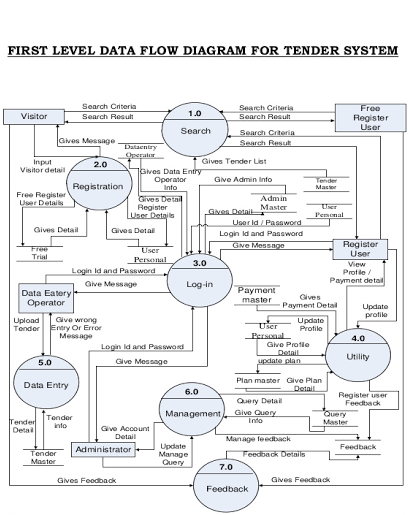
**4.3.2 FIRST LEVEL DATA FLOW DIAGRAM**

Figure 5: First level Data Flow Diagram for Tender Management System

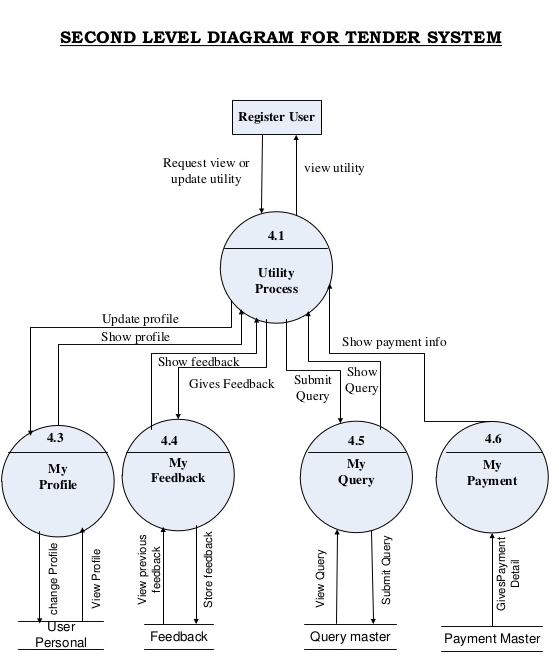
**4.3.3 SECOND LEVEL DATA FLOW DIAGRAM**

Figure 6: Second Level Data Flow Diagram for Tender Management System

**4.4 DATA MODELING**

**DATA STRUCTURE   REQUIREMENTS ACCORDING TO THE MODULES:**

* Identify the various tables required: Vendor, Tender, Bidder, Notice Table
* Fields for these tables: Vendor-Id, Tender-Id, Vendor-Name, Vendor-Info, etc.
* The Various Key Fields (for example Vendor-Id and Tender-Id are Primary key and foreign key).
* Various Constraints like Vendor-Id and Tender-Id can be Not Null, Unique etc.

**Describing Data Elements:**

Each entry in the data dictionary consists of a set of details describing the data used or produced in the system.  Each item is identified by a data name, description, alias, and length and has specific values that are permissible for it in the system being studied.

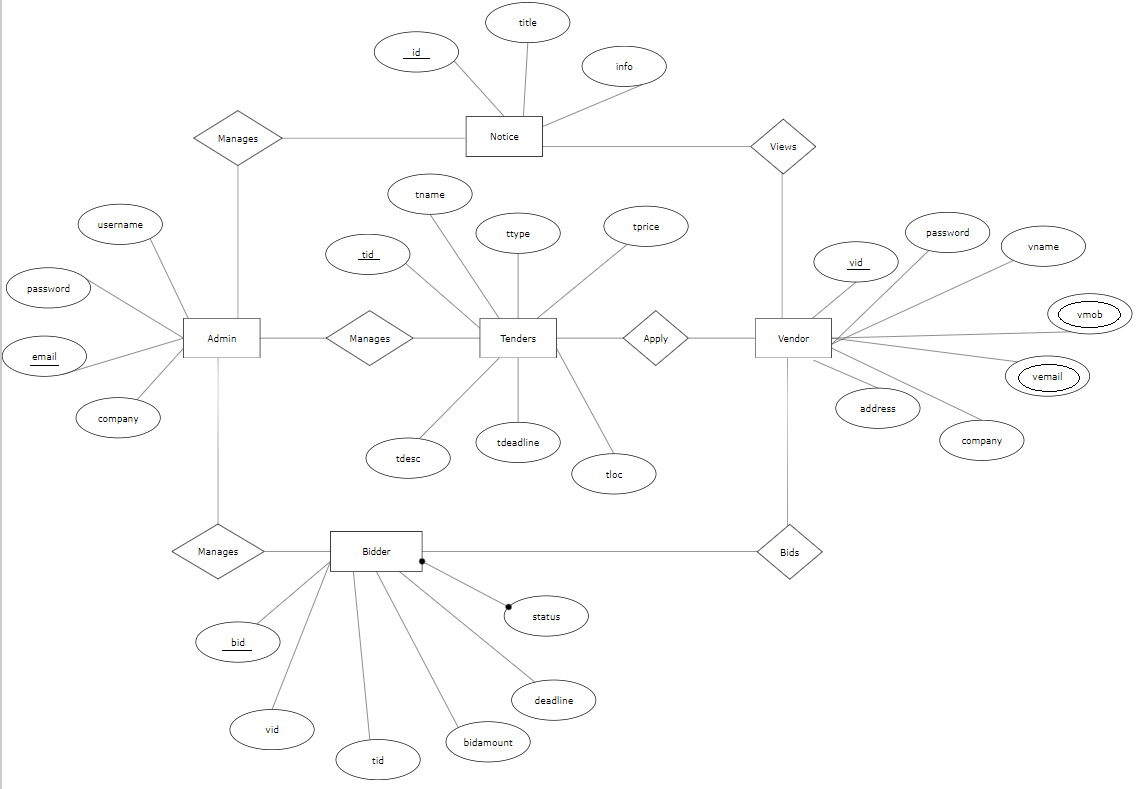
**4.5 E R DIAGRAM FOR TENDER MANAGEMENT SYSTEM**

Figure 7: Tender Management System Entity Relationship Diagram

**4.6 LIST OF TABLES:**

1. **Admin**
2. **Vendor**
3. **Bidder**
4. **Tender**
5. **Notice**
6. **Tender-Status**

**The Table Structure are given below:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Admin | | | | |
| Field Name | **Data Type** | **Size** | **Null** | **KEY** |
| company | varchar | 15 | No |  |
| email | varchar | 20 | Yes | Primary Key |
| username | varchar | 30 | Yes |  |
| password | int | 12 | Yes |  |

**Table 1: Admin Database Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Vendor | | | | |
| Field Name | **Data Type** | **Size** | **Null** | **KEY** |
| vid | **varchar** | **15** | **No** | **Primary Key** |
| password | **varchar** | **20** | **Yes** |  |
| vname | **varchar** | **30** | **Yes** |  |
| vmob | **int** | **12** | **Yes** |  |
| vemail | **varchar** | **40** | **Yes** |  |
| company | **varchar** | **20** | **Yes** |  |
| address | **varchar** | **100** | **Yes** |  |

**Table 2: Vendor Database Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Bidder | | | | |
| Field Name | **Data Type** | **Size** | **Null** | **KEY** |
| bid | **varchar** | **15** | **No** | **Primary Key** |
| vid | **varchar** | **15** | **Yes** |  |
| tid | **varchar** | **15** | **Yes** |  |
| bidamount | **int** | **11** | **Yes** |  |
| deadline | **date** |  | **Yes** |  |
| status | **varchar** | **10** | **Yes** |  |

**Table 3: Bidder Database Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Tender | | | | |
| Field Name | **Data Type** | **Size** | **Null** | **KEY** |
| tid | **varchar** | **15** | **No** | **Primary Key** |
| tname | **varchar** | **40** | **Yes** |  |
| ttype | **varchar** | **20** | **Yes** |  |
| tprice | **int** | **11** | **Yes** |  |
| tdesc | **varchar** | **300** | **Yes** |  |
| tdeadline | **date** |  | **Yes** |  |

**Table 4: Tender Database Table Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Notice | | | | |
| Field Name | **Data Type** | **Size** | **Null** | **KEY** |
| id | **int** | **3** | **No** | **Primary Key** |
| title | **varchar** | **35** | **Yes** |  |
| info | **varchar** | **300** | **Yes** |  |

**Table 5: Notice Database Table Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Tenderstatus | | | | |
| Field Name | **Data Type** | **Size** | **Null** | **KEY** |
| tender-id | **varchar** | **15** | **No** | **Primary Key** |
| bidder-id | **varchar** | **15** | **Yes** |  |
| status | **varchar** | **15** | **No** |  |
| vendor-id | **varchar** | **15** | **Yes** |  |

**Table 6: TenderStatus Database Table Structure**

**4.7 MODULE OF THE PROJECT**

MODULE 1: New Vendor Registration Module

MODULE 2: Authentication Module

MODULE 3: User Profile Module

MODULE 4: Tender Module

MODULE 5: Bidder Module

MODULE 6: Tender Status Module

MODULE 7: Add Tender Module

MODULE 8: Update or Remove Tender Module

MODULE 9: Tender Bid Acceptance or Rejection Module

MODULE 10: Notice Board Module

**4.8 PROCESS LOGIC FOR EACH MODULE**

**MODULE 1: VENDOR REGISTRATION MODULE**

In this module the vendors who are new to our site can register. During the registration process they need to provide some contact details and their information for further uses. As without registration no user can view any tenders available and they cannot bid for any tenders, so their registration is compulsory for utilizing this feature in our site.

**MODULE 2: AUTHENTICATION**

Whenever someone try to login into the website then their login credentials are checked according to the authentication module and if their login credentials entered are wrong then it may deny them to login and use the internal functionalities. The admin also have their login-id and password and they also need to enter correct details for successful login.

**MODULE 3: USER PROFILE**

After successful login of the vendor or user the user have different sections. One of those section is “Account” section. In the account section they may view their profile detail and if they want to change their details then they can click on update profile section and edit their details.

The user can also change the previous password and update it to new one for that they need to re-enter the old password.

**MODULE 4:** **TENDER MODULE**

In the tender module section, the users will be able to view all the tenders which are available to bid and also, they will see the tenders which gets already expired and has been assigned to some contractors. The Admin also can view, update, delete and add new tenders and launch it.

**MODULE 5:** **BIDDER MODULE**

If some vendor or contractor wants to work on a tender and interested in a particular tender then he needs to open the tender view section and go to bid module. In that module they can get access to bid for unassigned tenders with an amount greater than the base price of the respective tender.

**MODULE 6: TENDER STATUS MODULE**

If some user has already bid for some tender, then he can see the status of their bid in the “Tender status” section. The admin will be able to view all the bids and also, they will accept or reject a bid. If the admin performs such action then that information will be shown in the account of the vendor. After the acceptance of any tender bid by admin the assigned product Id and User Id will be shown in the notice section automatically.

**MODULE 7: ADD TENDER MODULE**

Only Admin have access to this module so whenever a company want to launch a tender then they may consult the admin and admin will add the tender details to the site. After the addition of the tender details the user will be able to see that tender and bid for that tender.

**MODULE 8: UPDATE OR REMOVE TENDER**

While addition of the tender, if some wrong details gets filled then admin have access to change and update the details by moving to update tender section. If the admin wants to remove any tender or stop any more bid for any tender then they may go to remove tender section and remove the tender from the public view section.

**MODULE 9: TENDER BID ACCEPTANCE OR REJECTION**

After bidding a tender the vendor or user can wait for their acceptance or rejection status. And this status can be seen in the section “Tender Bid Status”. But this approval and rejection is performed by the company admin. The admin will go to view the tender bids and then they may accept or reject the bids at that page.

**MODULE 10: NOTICE BOARD**

There is a notice section in the left side of the home screen which shows some notice regarding the projects, tenders, tender assignment status and other notice. This notice board is being added and updated by the company admin. In the notice module they may add, update or remove some notices from the notice board.

**4.9 USER INTERFACE DESIGN**

User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in a website, or technological product. This could be the control buttons of a radio, or the visual layout of a webpage.

We have used the CSS and JavaScript framework named Bootstrap for the front End Design and making the front-end user friendly.

**4.10 SECURITY ISSUES**

**SECURITY AND VALIDATION CHECKS**

In this project we have used following validation checks.

* While entering the data into the form it will check for the nullity of the fields in both of client machine and server.
* While entering the data the validation will be checking the existing records, if the records found a message will be fired to the end user as “The record already exists”.
* The vendors are being granted with access only to the vendor details and bidding of tender and access to their own account.
* Passwords are encrypted before storing the data into the database. And it will be decrypted and authenticated while login into the system.
* The Vendor login and logout status are tracked using the Session Tracking in Java. If someone logout then no one will have access or open their account without correct login details.

**CHAPTER: 5**

**SYSTEM DEVELOPMENT**

**5.1 IMPLEMENTATION APPROACHES**

This web project can be deployed on Tomcat Web Server from the Express Edition of the Eclipse IDE.

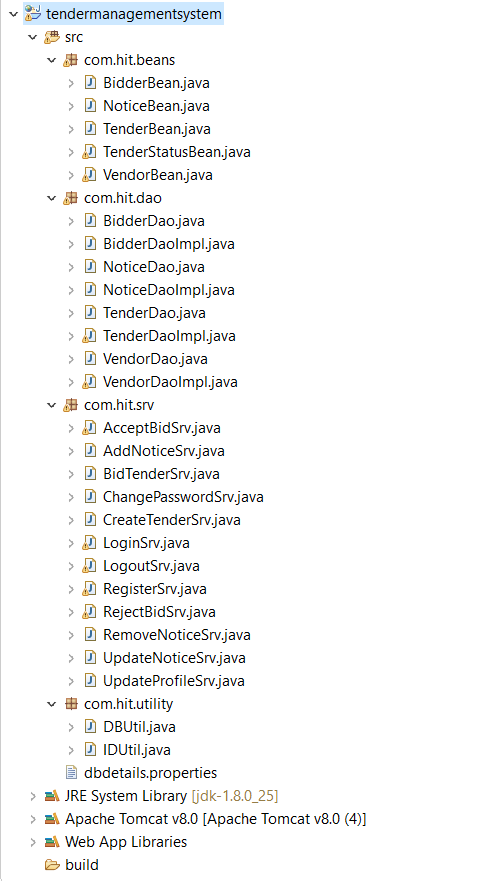
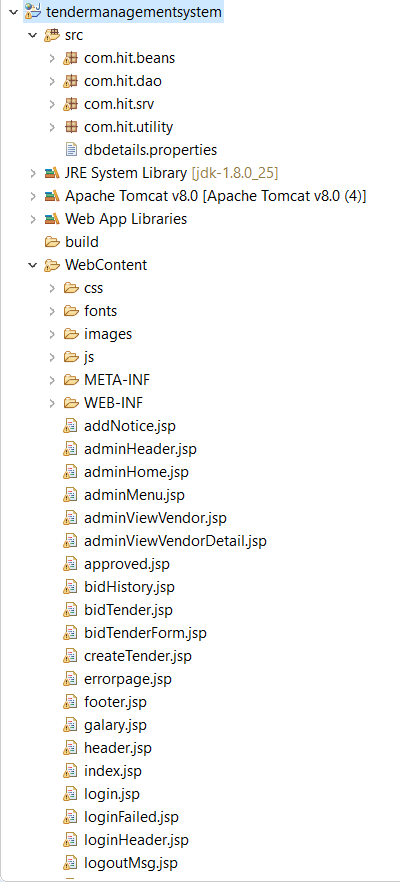
The Directory Structure of the Project is Assigned here:****

Figure 8: Project Directory Structure

**CHAPTER: 6**

**TESTING AND OUTPUTS**

**6.1 TESTING**

The different types of testing are as follows

 1. **Unit Testing:**

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

1. **Integration Testing:**

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

1. **System Testing:**

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

1. **Field Testing:**

Field testing is a critical step in the recapitulation cycle, helping you find out the flaws in your offering. The purpose of the field testing is to minimize risk, by making sure that the offering works beforehand you deliver it to the customers.

1. **Acceptance Testing:**

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

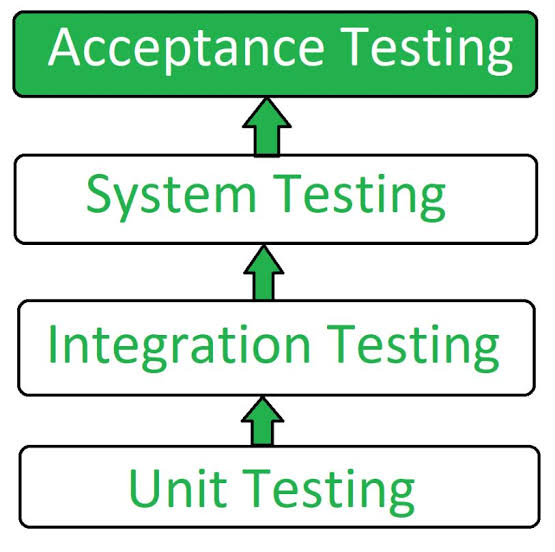
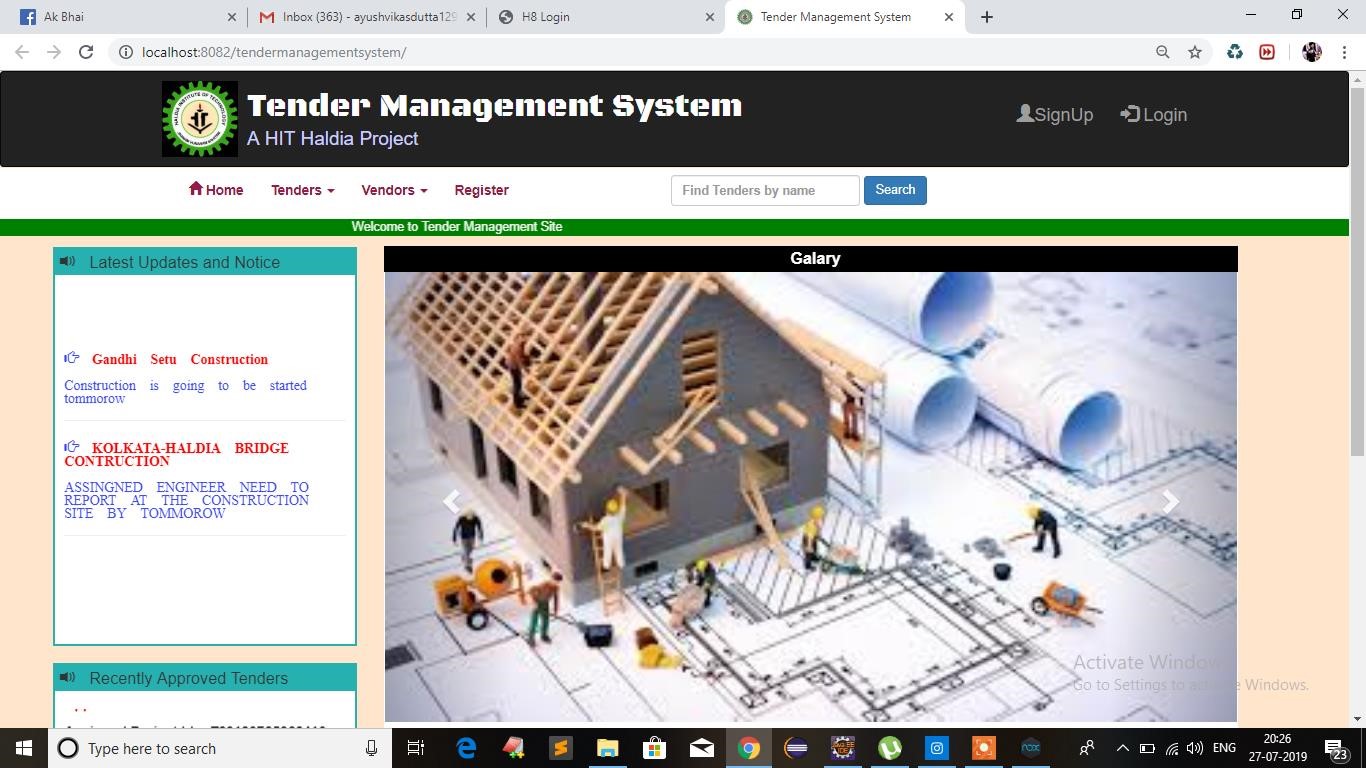


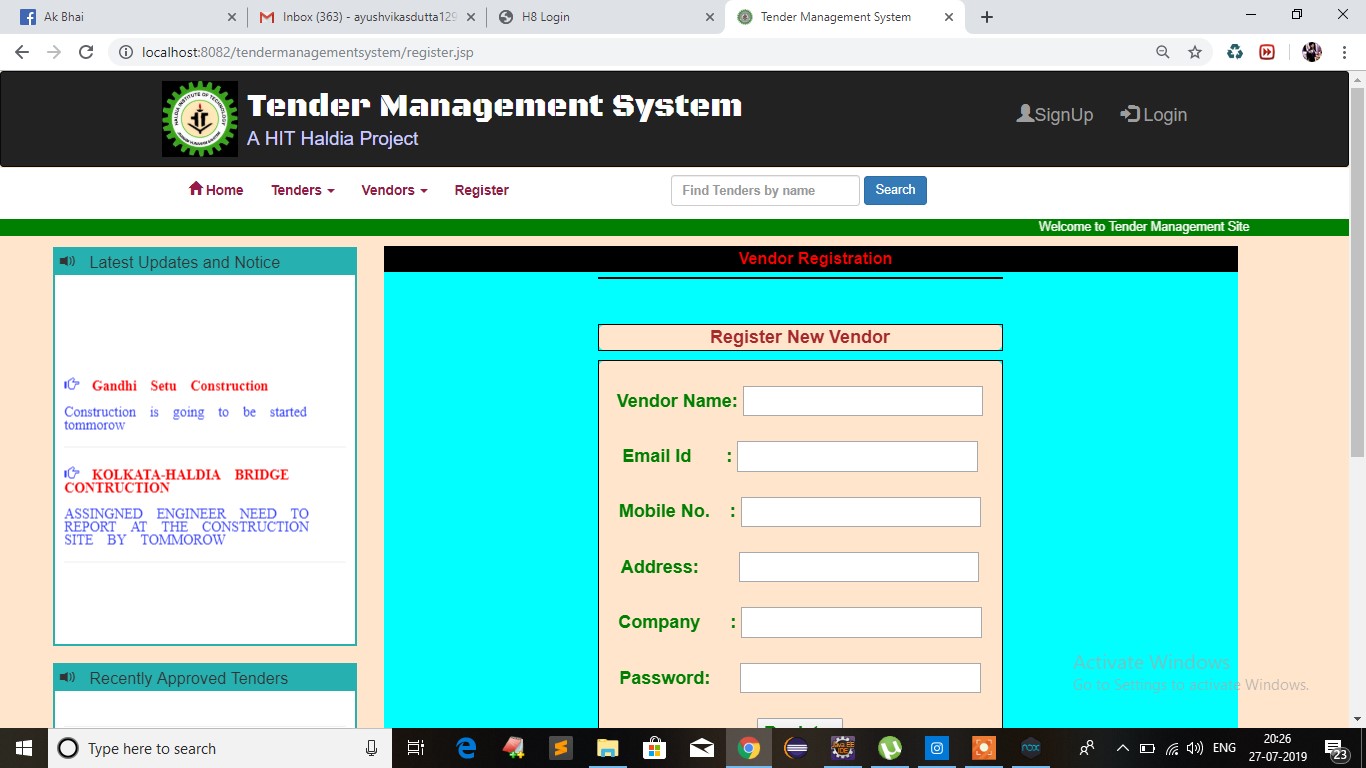
Figure 9: Types of Testing

**6.2 OUTPUT SCREENSHOTS**

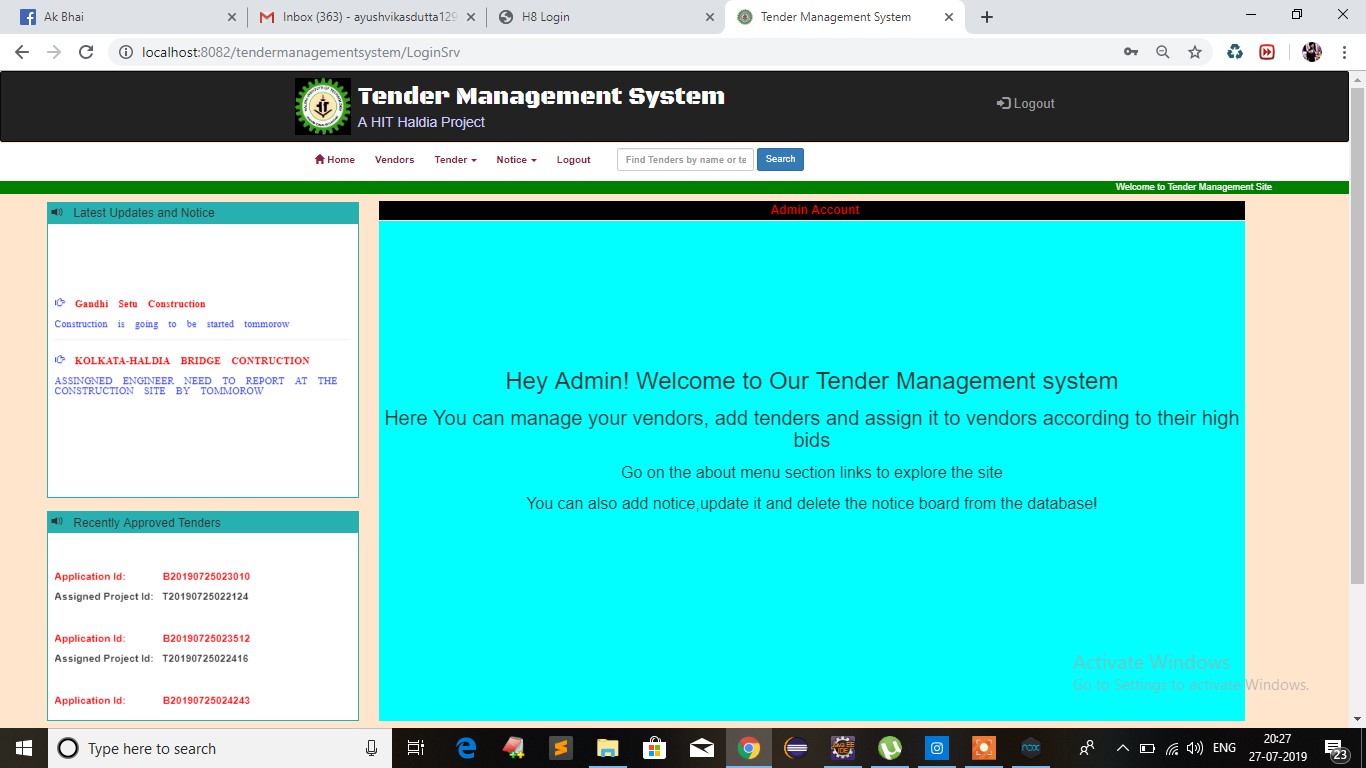
**HOME PAGE**



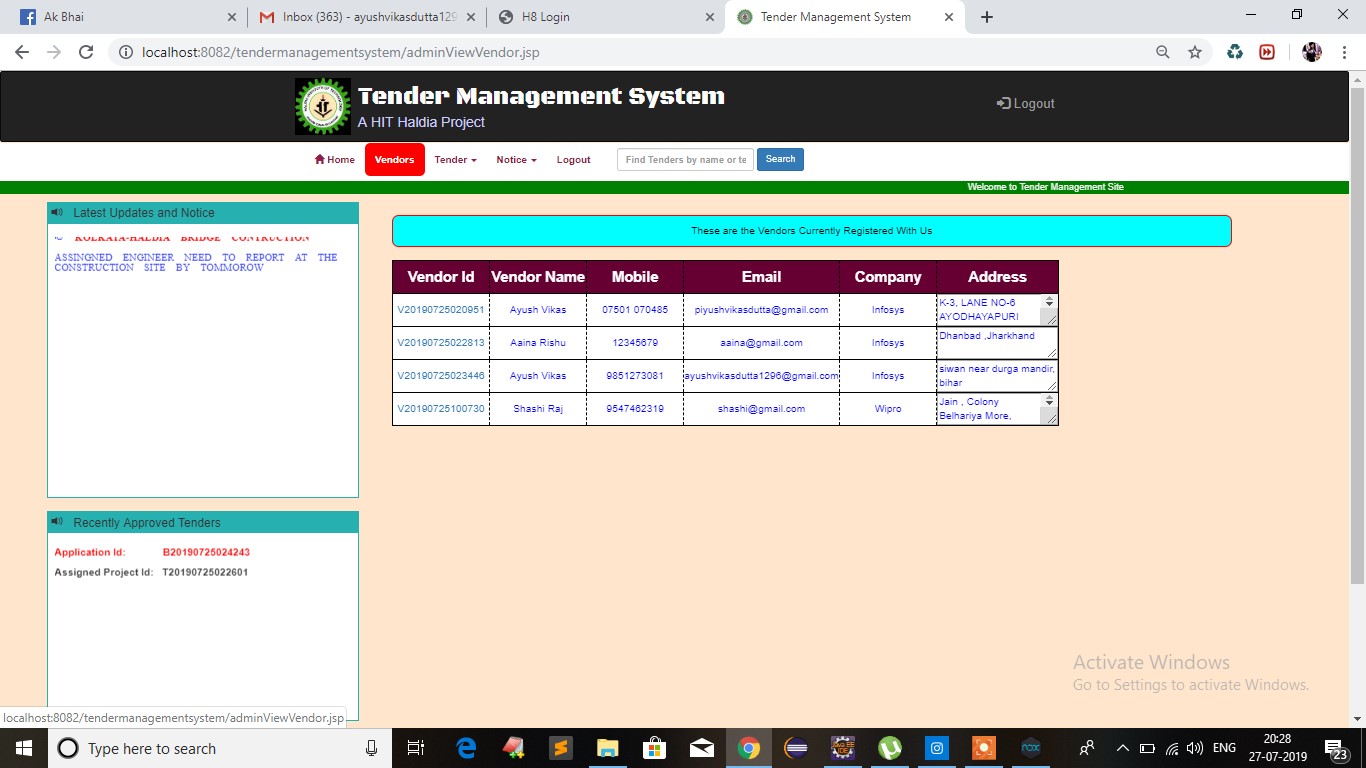
**Figure 10: Screenshot of Home Page**

**NEW VENDOR REGISTER**

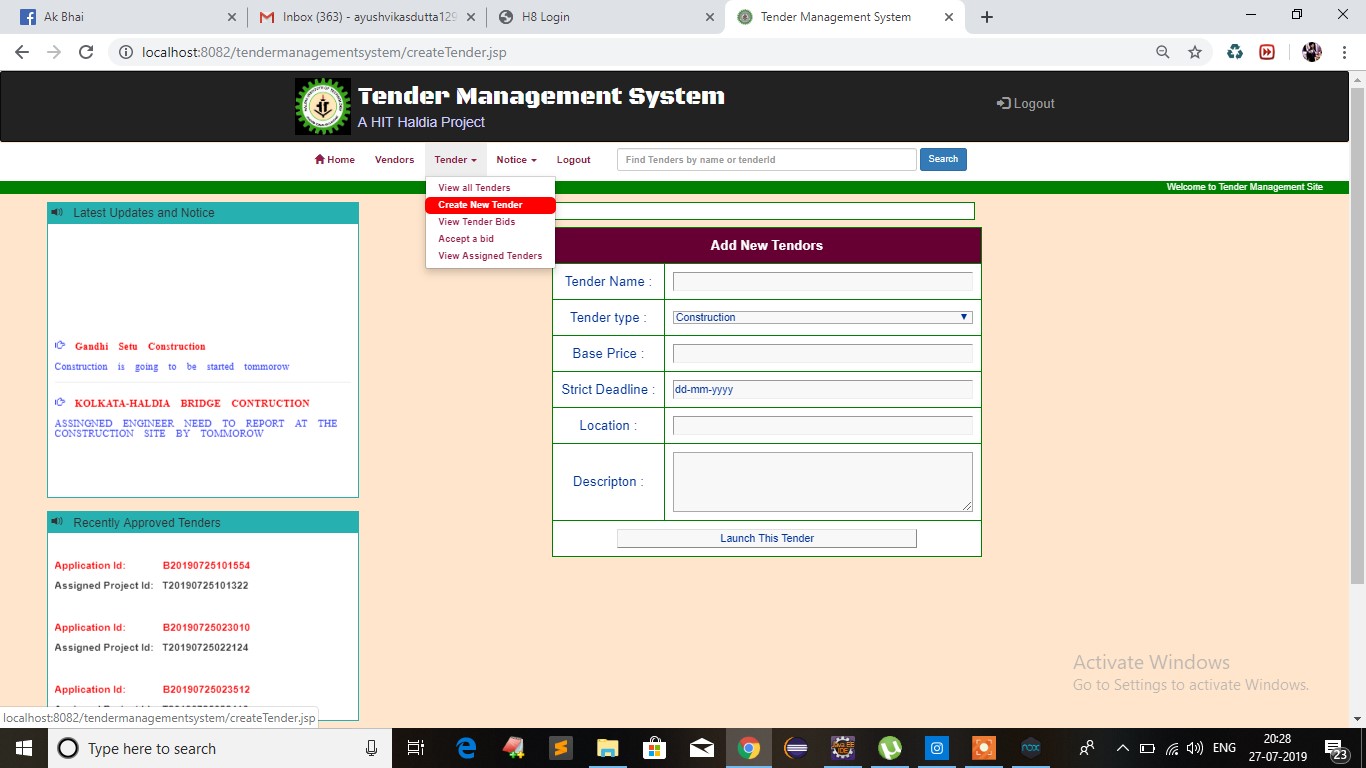
**Figure 11: Screenshot of Vendor Registration**

**ADMIN HOME PAGE**

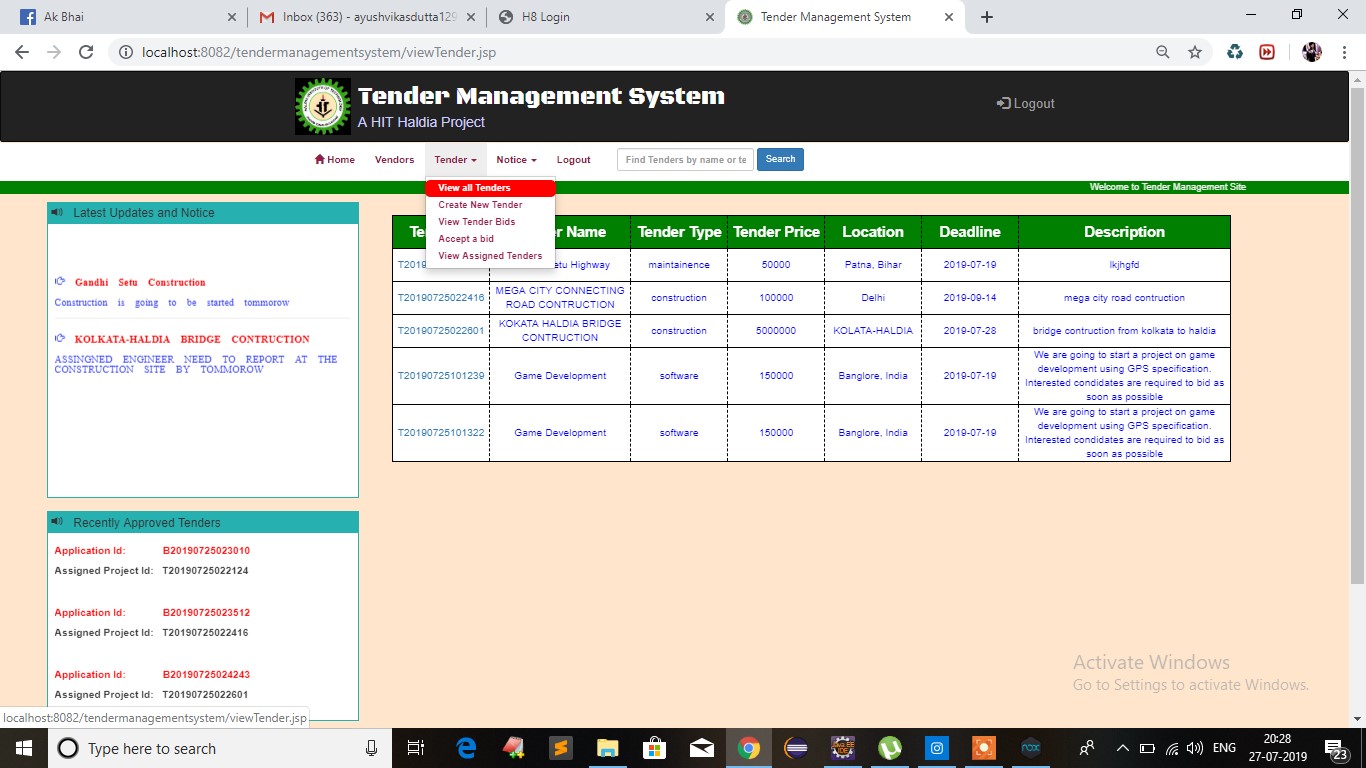
**Figure 12: Screenshot of Admin Home Page**

**VIEW VENDER DETAILS**

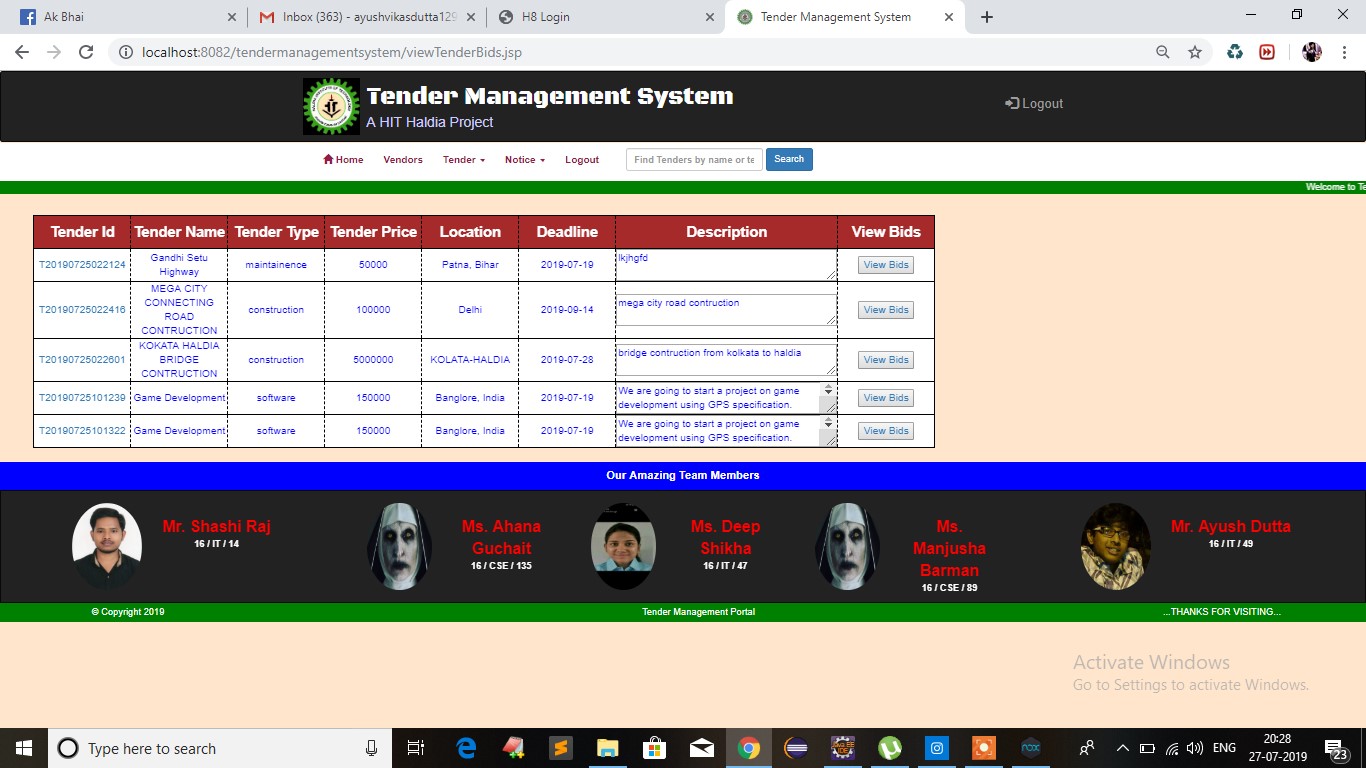
**Figure 13: Screenshot of View Vendor Page**

**VIEW TENDERS**

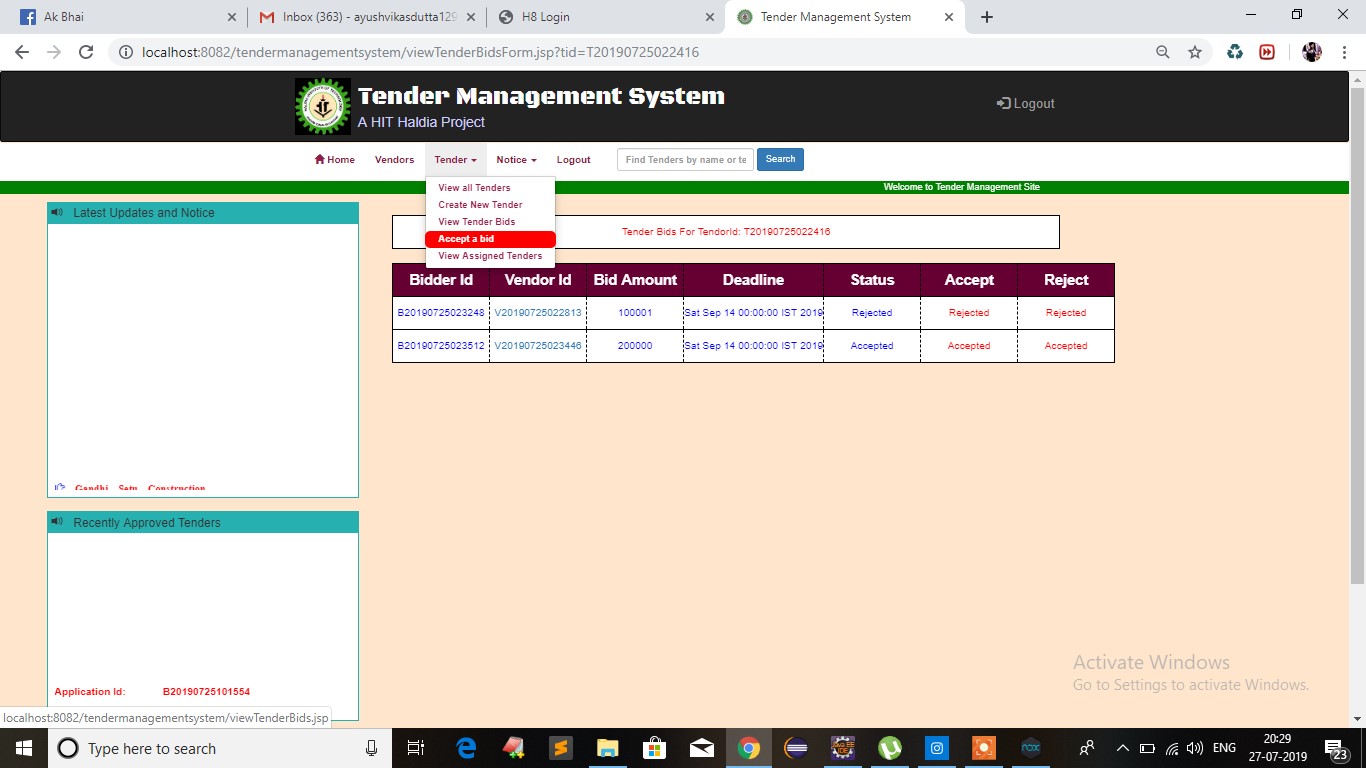
**Figure 14: Screenshot of vendor list view by admin**

**ADD NEW TENDERS**

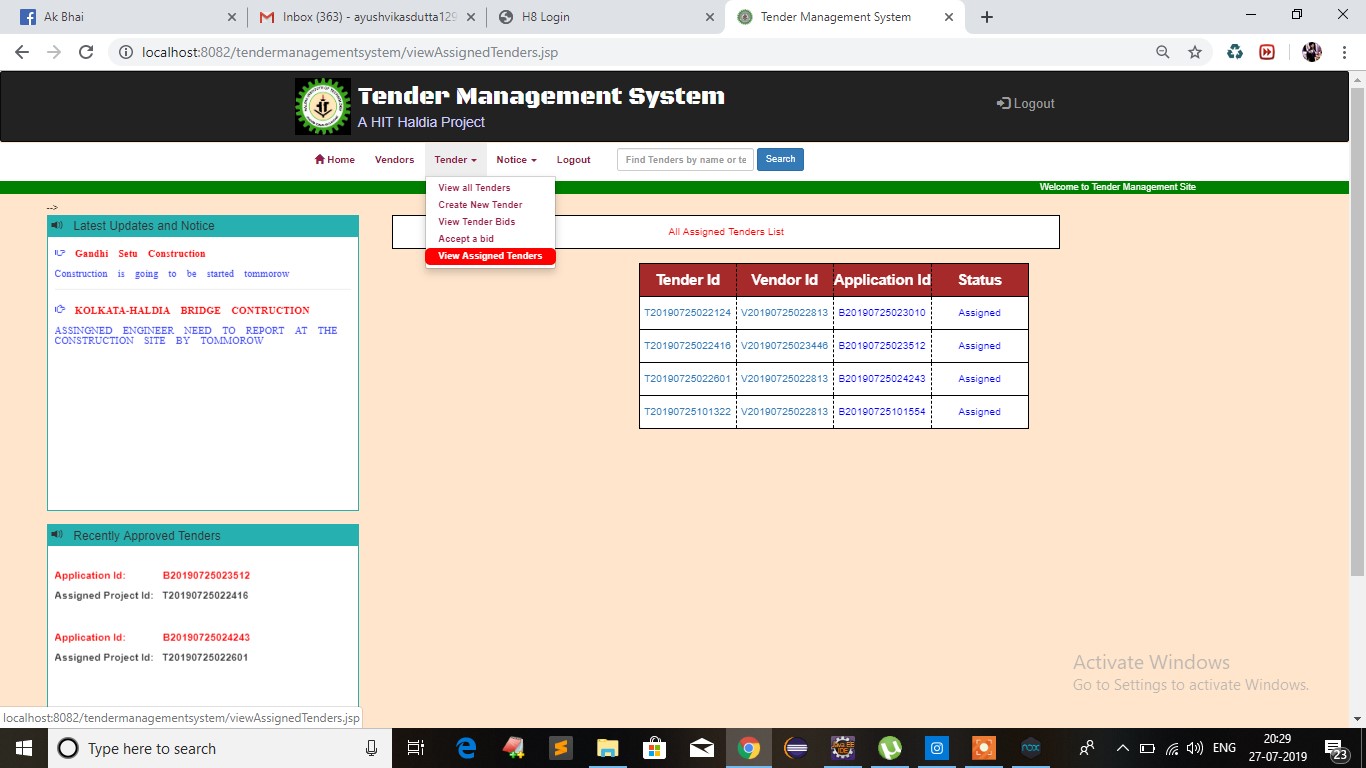
**Figure 15: Screenshot of Add New Tender Page**

**VIEW TENDER BIDS**

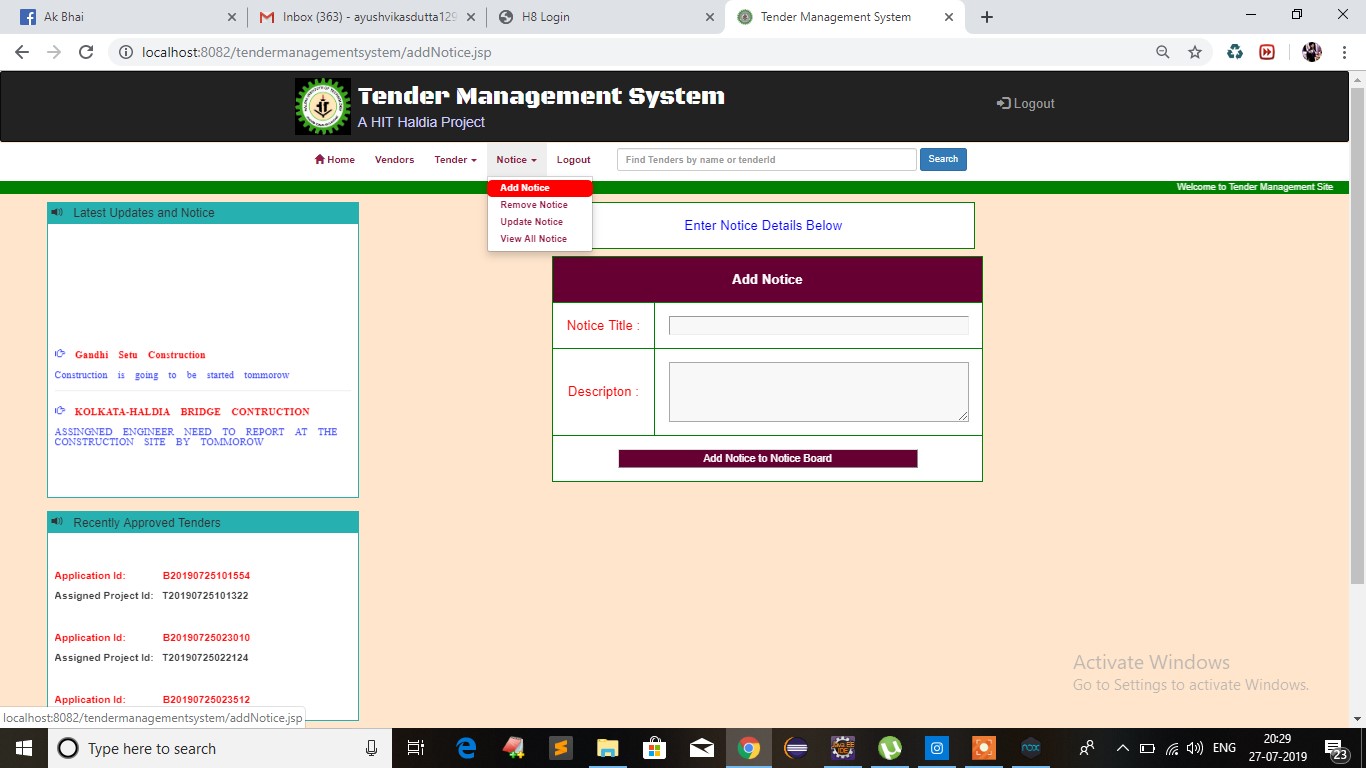
**Figure 16: Screenshot of Tender Bids View Page**

**ACCEPT OR REJECT TENDER**

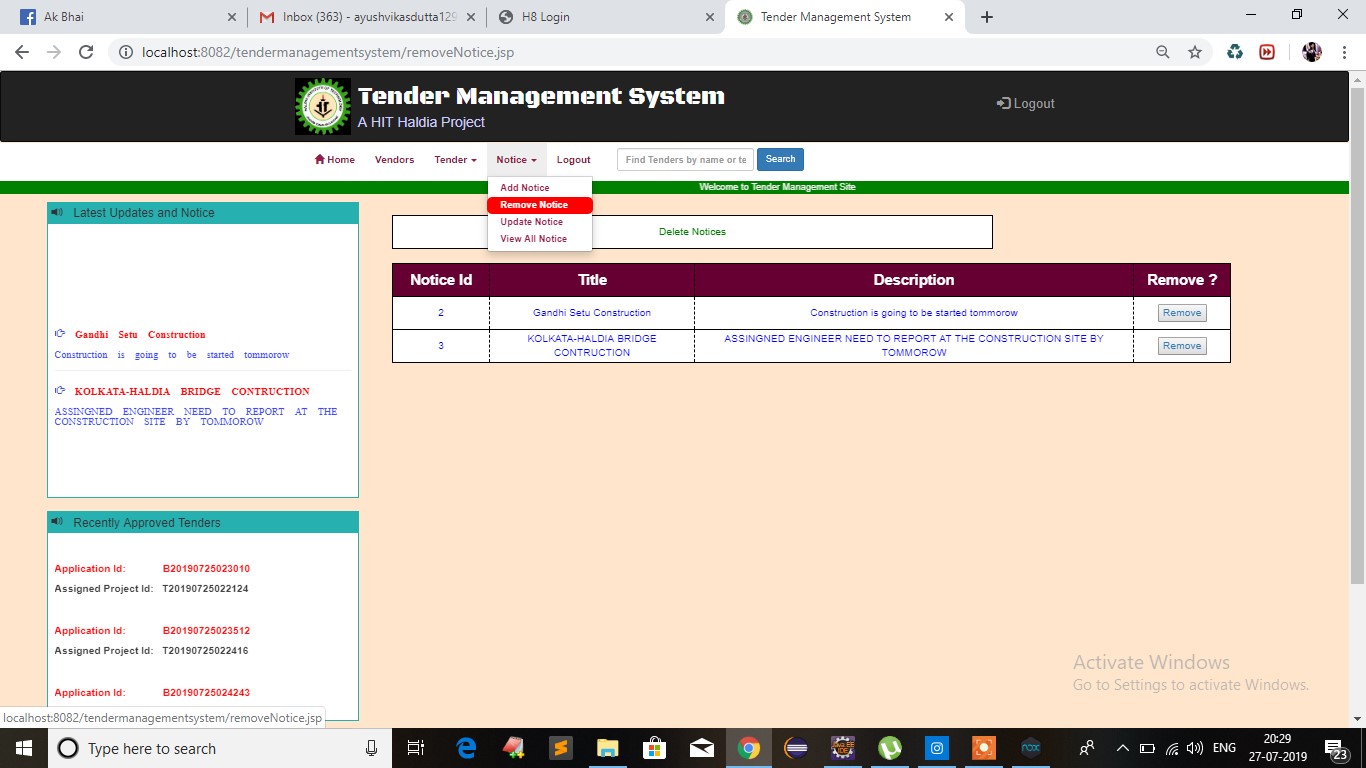
**Figure 17: Screenshot of Tender Reject or Acceptance Page**

**ASSIGNED TENDERS**

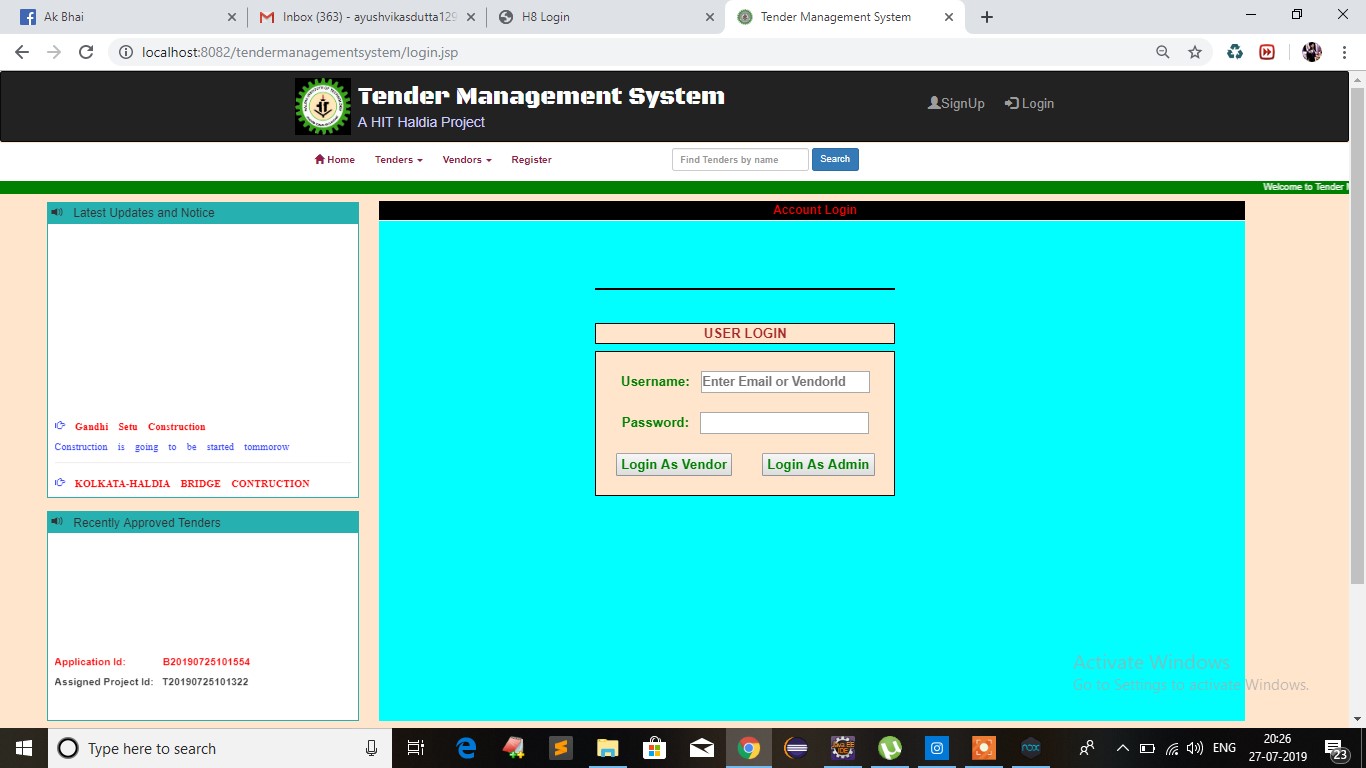
**Figure 18: Screenshot of Assigned Tenders list Page**

**ADD NOTICE**

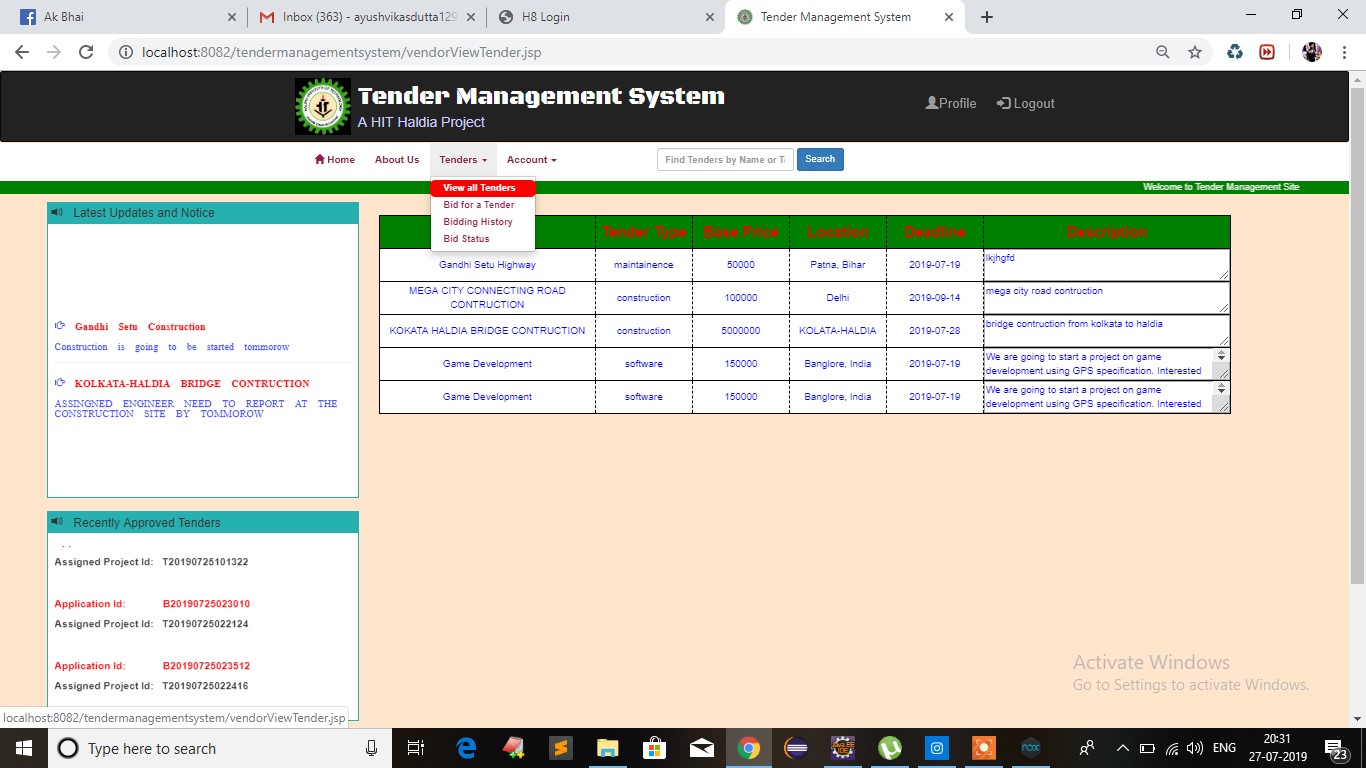
**Figure 19: Screenshot of Add Notice Page**

**REMOVE NOTICE**

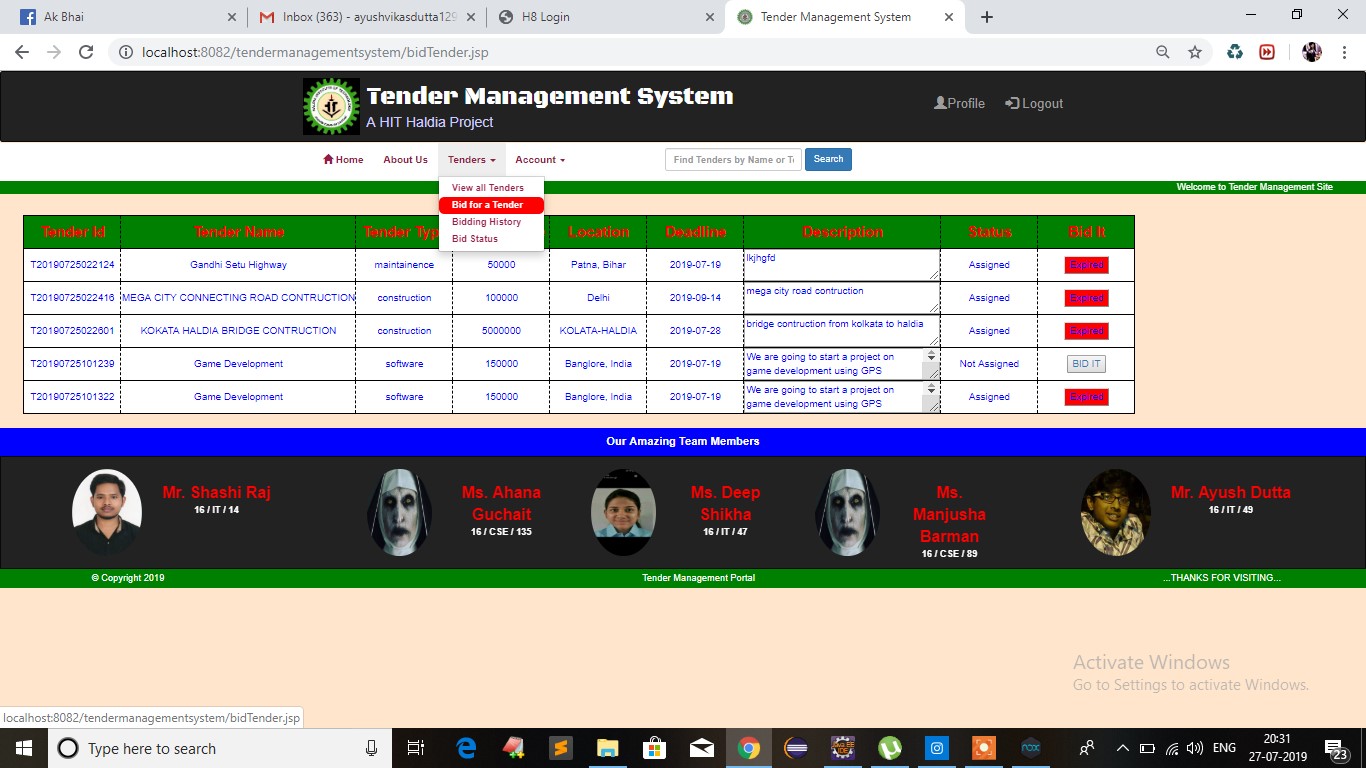
**Figure 20: Screenshot of Notice Removal Page**

**LOGIN AS VENDOR**

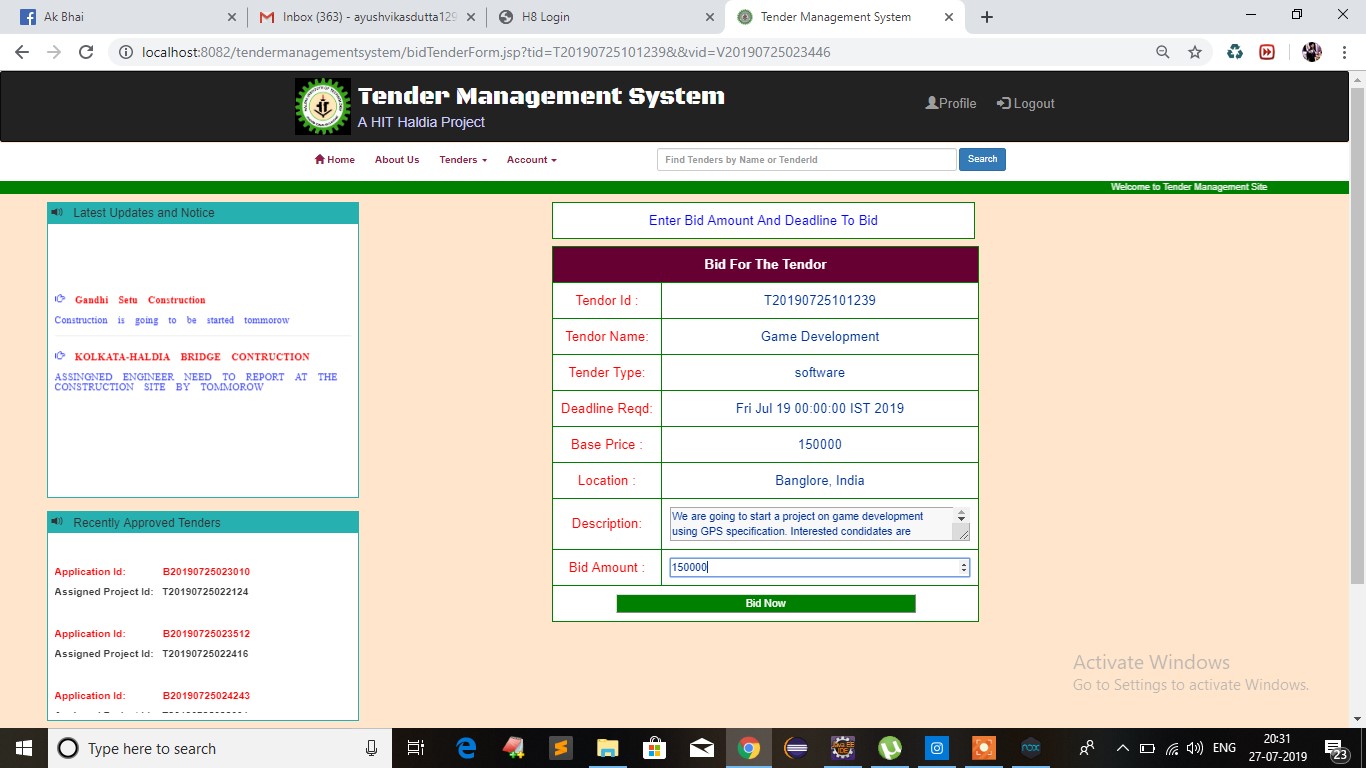
**Figure 21: Screenshot of Vendor Login Page**

**VIEW RELEASED TENDERS**

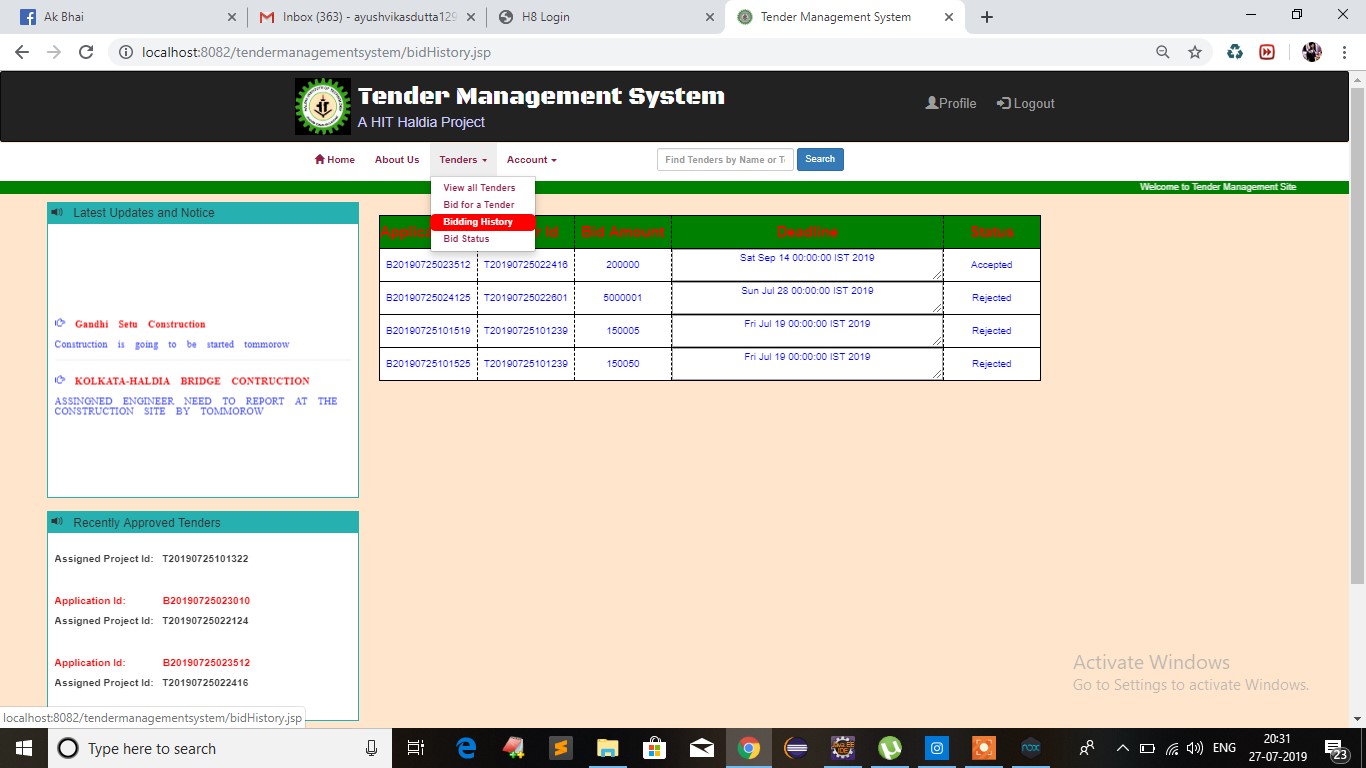
**Figure 22: Screenshot of released Tender list Page**

**BID** **FOR A TENDER**

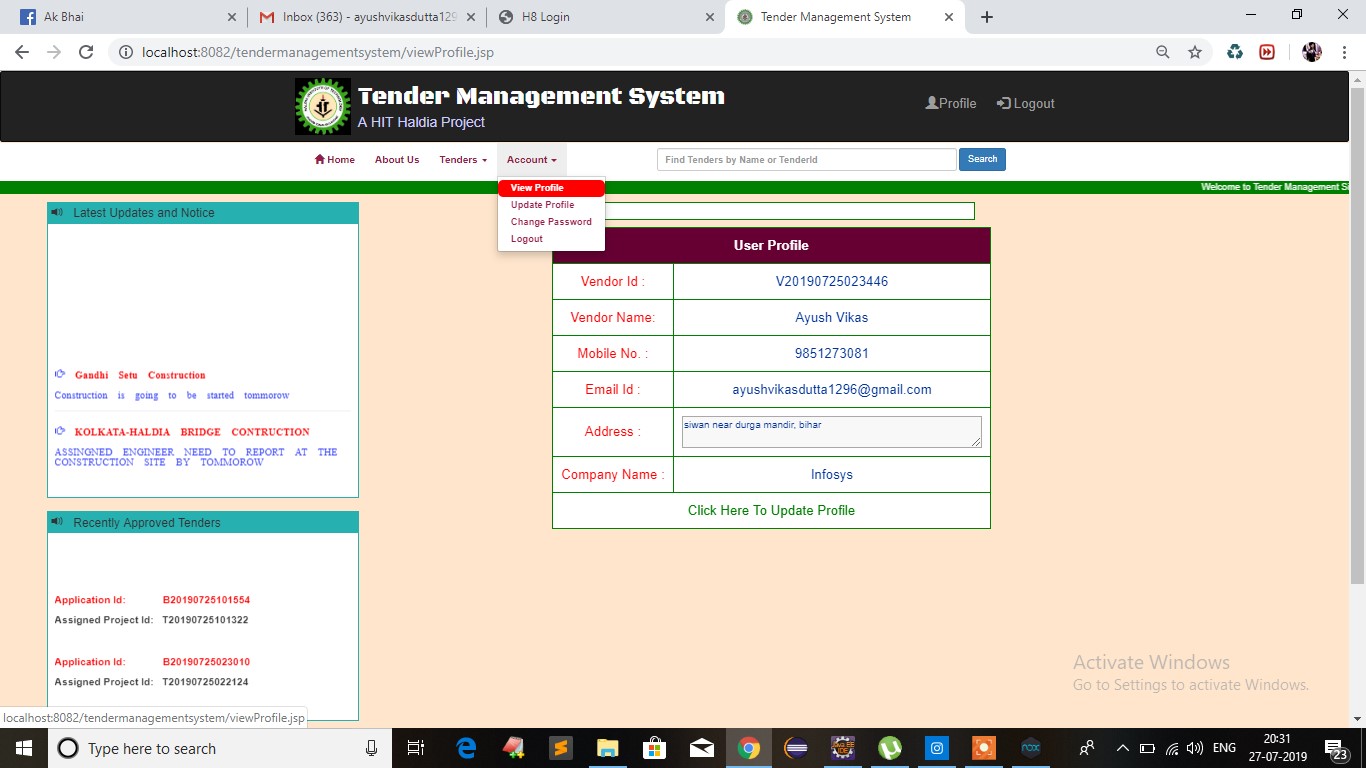
**Figure 23: Screenshot of Bidding Tenders Page**

**UDPATE TENDER DETAILS**

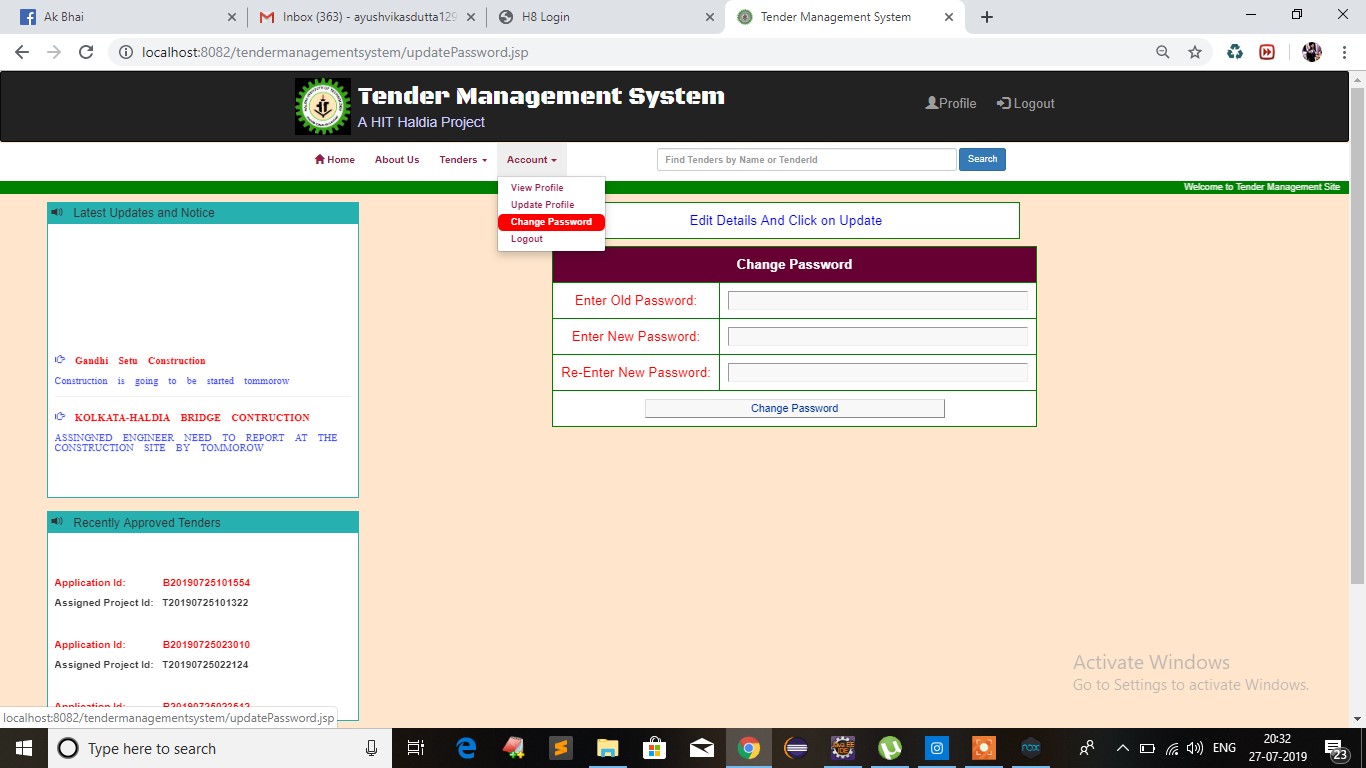
**Figure 24: Screenshot of Update Tender Page**

**TENDER BID STATUS**

**Figure 25: Screenshot of Bidding Status Page**

**VIEW PROFILE**

**Figure 26: Screenshot of Vendors Profile Page**

**CHANGE PASSWORD**

**Figure 27: Screenshot of Vendor Password Change Module**

**CHAPTER: 7**

**CONCLUSION**

Our project Tender Management System has successfully created a user-friendly environment in managing tenders online. our project is flexible and changes can be incorporated easily. It is so designed that further modifications to the system can be easily done.

The application illustrated or demonstrates the way to develop an online Travel Deal by using interactive web client by using JSP, Servlet with safer way to access &encapsulate database by EJB component. This suggests the application server simply deployable and accessible.

The project ensures the user to get benefitted by the Technology, so that they can enjoy the services with their own bidding by sitting at their residence. They can even look for the approval status and also get the project notice from their home, also the company saves a lot of time and money.

* 1. **LIMITATIONS OF THE SYSTEM**

**1. Security:**

The biggest drawback of e-commerce is the issue of security. People fear to provide personal and financial information, even though several improvements have been made in relation to data encryption. Certain websites do not have capabilities to conduct authentic transactions. Fear of providing credit card information and risk of identity limit the growth of online tender management system.

1. **Constructors Stipend:**

As the projects are assigned according to the bid amount so the contractor bidding the lowest amount would have greater chance of getting the project to be assigned. So the vendor needs to work for the lower stipend.

**CHAPTER: 8**

**FUTURE SCOPE OF THE PROJECT**

The deliverable product is named as Online Tender Management System. I have designed the given proposed Website in Java to automate the tender management System.

In the era of digitalization, the horizon of business opportunities is expanding like never before. The bidding ecosystem is now invigorated with the digital transformation. Online tendering systems are going to eliminate all the issues of the traditional tendering process and make it a seamless exercise. Also, this platform will enable bidders to bid for International tenders. Whether they are Nepal tenders or tenders for Australia or Brazil, bidders can have easy access to tenders across the world.

The utilization of this online tender management system may lead to improve the company profits and help in the company growth. The acceptability of Online Tendering has been on a rise in recent times. This is because more and more organizations are realizing the potential of Online Tendering in curbing irregularities in supply chain and streamlining the whole procurement process. Also, there are instances of organizations reaping huge benefits as a result of successful implementation of Online Tendering Management platforms. This fact has enthused many other organizations to follow suit.

The relevance of Online Tendering is poised to grow as more and more enterprises and organizations adopt e-Tendering and start reaping its benefits. Government has by now realized that Online Tendering is indispensable for a hassle-free and streamlined procurement process. Online Tendering would soon become an integral part of supply chain management of any organization, given the range of benefits it provides. Tenders floated by Governmental organizations would be increasingly executed

**BIBLIOGRAPHY**

[1] <https://www.tutorialspoint.com/jdbc/index.html>

[2] <https://www.dummies.com/programming/java/what-is-java-and-why-is-it-so-great/>

[3] <https://www.javatpoint.com/jsp-tutorial>

[4] <https://www.geeksforgeeks.org/introduction-java-servlets/>

[5] <https://www.executivecompass.co.uk/blog/bid-management/e-tendering/>

[6] <https://www.purchasing-procurement-center.com/tender-management-system.html>

[7] <https://blog.aufaittechnologies.com/types-of-tenders-and-tendering-process/>

[8] <https://www.slideshare.net/smitpatel10192/etender-system>

[9] <https://www.techbeamers.com/testing-types/>

[10] <https://docs.oracle.com/javase/tutorial/>

[11] <https://en.wikipedia.org/wiki/Data-flow_diagram>

[12] <https://dev.mysql.com/doc/mysql-getting-started/en/>

[13] <https://www.theserverside.com/definition/J2EE-Java-2-Platform-Enterprise-Edition>

[14] <https://www.w3schools.com/html/default.asp>

[15] <https://getbootstrap.com/docs/4.3/getting-started/introduction/>

[16] <https://www.c1india.com/how-mandating-e-tendering-helps/>

**SAMPLE CODE**

**TenderDao.java**

package com.hit.dao;

import java.util.List;

import com.hit.beans.TenderBean;

import com.hit.beans.TenderStatusBean;

public interface TenderDao {

public List<TenderBean> getTenderDetails(String id);

public List<TenderBean> getAllTenders();

public String createTender(TenderBean tender);

public boolean removeTender(String tid);

public String updateTender(TenderBean tender);

public TenderBean getTenderDataById(String tenderId);

public String getTenderStatus(String tenderId);

public String assignTender(String tenderId,String vendorId,String bidderId);

public List<TenderStatusBean> getAllAssignedTenders();

}

**VendorDao.java**

package com.hit.dao;

import java.util.List;

import com.hit.beans.VendorBean;

public interface VendorDao {

public String registerVendor(VendorBean vendor);

public List<VendorBean> getAllVendors();

public boolean validatePassword(String vendorId,String password);

public String updateProfile(VendorBean vendor);

public String changePassword(String vendorId,String oldPassword,String newPassword);

public VendorBean getVendorDataById(String vendorId);

}

**NoticeDao.java**

package com.hit.dao;

import java.util.List;

import com.hit.beans.NoticeBean;

public interface NoticeDao {

public String removeNotice(int noticeId);

public String addNotice(String noticeTitle, String noticeDesc );

public List<NoticeBean> viewAllNotice();

public String updateNotice(NoticeBean notice);

public NoticeBean getNoticeById(int noticeId);

}

**BidderDao.java**

package com.hit.dao;

import java.util.List;

import com.hit.beans.BidderBean;

public interface BidderDao {

public String acceptBid(String applicationId,String tenderId,String vendorId);

public String rejectBid(String applicationId);

public String bidTender(String tenderId, String vendorId,String bidAmount,String deadline);

public List<BidderBean> getAllBidsOfaTender(String tenderId);

public List<BidderBean> getAllBidsOfaVendor(String vendorId);

}

**BidderDaoImpl.java**

package com.hit.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import com.hit.beans.BidderBean;

import com.hit.utility.DBUtil;

import com.hit.utility.IDUtil;

public class BidderDaoImpl implements BidderDao{

@Override

public String acceptBid(String applicationId,String tenderId,String vendorId) {

String status = "Bid Acceptance Failed";

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;

PreparedStatement pst = null;

ResultSet rs = null;

try {

ps = con.prepareStatement("select \* from tenderstatus where tid=?");

ps.setString(1, tenderId);

rs = ps.executeQuery();

if(rs.next()){

status = "Project Already Assigned";

}

else{

pst = con.prepareStatement("update bidder set status = ? where bid=? and status=?" pst.setString(1, "Accepted");

pst.setString(2, applicationId);

pst.setString(3, "Pending");

int x = pst.executeUpdate();

if(x>0){

status = "Bid Has Been Accepted Successfully!";

TenderDao dao = new TenderDaoImpl();

status = status + "<br>"+dao.assignTender(tenderId, vendorId,applicationId);

}

}

} catch (SQLException e) {

status = status + "Error: "+e.getMessage();

e.printStackTrace();

}

finally{

DBUtil.closeConnection(con);

DBUtil.closeConnection(ps);

}

return status;

}

@Override

public String rejectBid(String applicationId) {

String status = "Bid Rejection Failed";

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;

try {

ps = con.prepareStatement("update bidder set status = ? where bid=? and status = ?");

ps.setString(1, "Rejected");

ps.setString(2, applicationId);

ps.setString(3, "Pending");

int x = ps.executeUpdate();

if(x>0)

status = "Bid Has Been Rejected Successfully!";

} catch (SQLException e) {

status = status + "Error: "+e.getMessage();

e.printStackTrace();

}

finally{

DBUtil.closeConnection(con);

DBUtil.closeConnection(ps);

}

return status;

}

@Override

public String bidTender(String tenderId, String vendorId, String bidAmount,

String bidDeadline) {

String status = "Tender Bidding Failed!";

String bidId = IDUtil.generateBidderId();

String bidStatus = "Pending";

BidderBean bidder = new BidderBean(bidId, vendorId, tenderId, bidAmount, bidDeadline, bidStatus);

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;

try {

ps = con.prepareStatement("insert into bidder values(?,?,?,?,?,?)");

ps.setString(1, bidId);

ps.setString(2, vendorId);

ps.setString(3,tenderId);

ps.setInt(4, bidder.getBidAmount());

java.sql.Date bidDate = new java.sql.Date(bidder.getBidDeadline().getTime());

ps.setDate(5, bidDate);

ps.setString(6, bidStatus);

int x = ps.executeUpdate();

if(x>0)

status = "You have successfully Bid for the tender";

} catch (SQLException e) {

e.printStackTrace();

}

finally{

DBUtil.closeConnection(con);

DBUtil.closeConnection(ps);

}

return status;

}

@Override

public List<BidderBean> getAllBidsOfaTender(String tenderId) {

List<BidderBean> bidderList = new ArrayList<BidderBean>();

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;

ResultSet rs = null;

try {

ps = con.prepareStatement("select \* from bidder where tid=?");

ps.setString(1, tenderId);

rs = ps.executeQuery();

while(rs.next()){

BidderBean bidder = new BidderBean();

bidder.setBidAmount(rs.getInt("bidamount"));

bidder.setBidDeadline(new java.sql.Date(rs.getDate("deadline").getTime()));

bidder.setBidId(rs.getString("bid"));

bidder.setBidStatus(rs.getString("status"));

bidder.setTenderId(rs.getString("tid"));

bidder.setVendorId(rs.getString("vid"));

bidderList.add(bidder);

}

} catch (SQLException e) {

e.printStackTrace();

}

finally{

DBUtil.closeConnection(con);

DBUtil.closeConnection(ps);

DBUtil.closeConnection(rs);

}

return bidderList;

}

@Override

public List<BidderBean> getAllBidsOfaVendor(String vendorId) {

List<BidderBean> bidderList = new ArrayList<BidderBean>();

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;

ResultSet rs = null;

try {

ps = con.prepareStatement("select \* from bidder where vid=?");

ps.setString(1, vendorId);

rs = ps.executeQuery();

while(rs.next()){

BidderBean bidder = new BidderBean();

bidder.setBidAmount(rs.getInt("bidamount"));

bidder.setBidDeadline(new java.sql.Date(rs.getDate("deadline").getTime()));

bidder.setBidId(rs.getString("bid"));

bidder.setBidStatus(rs.getString("status"));

bidder.setTenderId(rs.getString("tid"));

bidder.setVendorId(rs.getString("vid"));

bidderList.add(bidder);

}

} catch (SQLException e) {

e.printStackTrace();

}

finally{

DBUtil.closeConnection(con);

DBUtil.closeConnection(ps);

DBUtil.closeConnection(rs);

}

return bidderList;

}

}

**DBUtil.java**

package com.hit.utility;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DBUtil {

private static Connection conn = null;

public DBUtil(){

}

public static Connection provideConnection(){

ResourceBundle rb = ResourceBundle.getBundle("dbdetails");

String cs = rb.getString("connectionString");

String username = rb.getString("username");

String password = rb.getString("password");

String dbDriver = rb.getString("driverName");

try {

if(conn == null || conn.isClosed()){

try {

Class.forName(dbDriver);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

conn = DriverManager.getConnection(cs,username,password);

}

}

catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return conn;

}

public static void closeConnection(PreparedStatement ps){

try {

if(ps != null)

ps.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void closeConnection(ResultSet rs){

try {

if(rs != null)

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void closeConnection(Connection con){

try {

if(con != null)

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

**IDUtil.java**

package com.hit.utility;

import java.text.SimpleDateFormat;

import java.util.Date;

public class IDUtil {

public static String generateTenderId(){

String tid = null;

SimpleDateFormat sdf=new SimpleDateFormat("yyyyMMddhhmmss");

tid=sdf.format(new Date());

tid = "T"+tid;

return tid;

}

public static String generateVendorId(){

String vid = null;

SimpleDateFormat sdf=new SimpleDateFormat("yyyyMMddhhmmss");

vid=sdf.format(new Date());

vid = "V"+vid;

return vid;

}

public static String generateApplicationId(){

return generateBidderId();

}

public static String generateBidderId(){

String bidderId = null;

SimpleDateFormat sdf=new SimpleDateFormat("yyyyMMddhhmmss");

bidderId =sdf.format(new Date());

bidderId = "B"+bidderId;

return bidderId;

}

}

**LoginServlet.java**

package com.hit.srv;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.hit.beans.VendorBean;

import com.hit.utility.DBUtil;

@WebServlet("/LoginSrv")

public class LoginSrv extends HttpServlet {

private static final long serialVersionUID = 1L;

public LoginSrv() {

super();

}

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

doPost(request,response);

}

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

response.setContentType("text/html");

//PrintWriter pw = response.getWriter();

String uname = request.getParameter("username").trim();

String pword = request.getParameter("password").trim();

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

if(user.toLowerCase().equals("login as admin")){

//Admin login check

if(uname.equals("Admin") && pword.equals("Admin")){

//login successful

HttpSession session = request.getSession();

session.setAttribute("user","admin");

session.setAttribute("username", uname);

session.setAttribute("password", pword);

RequestDispatcher rd = request.getRequestDispatcher("adminHome.jsp");

rd.forward(request, response);

}

else{

PrintWriter pw = response.getWriter();

RequestDispatcher rd = request.getRequestDispatcher("login.jsp");

rd.include(request, response);

pw.print("<script>document.getElementById('show').innerHTML = 'Invalid Username or Password!!'</script>");

}

}

else if(user.toLowerCase().equals("login as vendor")){

//Vendor Login Check

Connection conn =DBUtil.provideConnection();

PreparedStatement ps = null;

PreparedStatement pst = null;

ResultSet rs = null;

ResultSet rs1 = null;

try {

pst = conn.prepareStatement("select \* from vendor where vid=? and password=?");

pst.setString(1, uname);

pst.setString(2, pword);

rs=pst.executeQuery();

if(rs.next()){ //Vendor Login Successful

HttpSession session = request.getSession();

session.setAttribute("user","user");

session.setAttribute("username", uname);

session.setAttribute("password", pword);

String vid = uname;

String pass = pword;

String vname = rs.getString("vname");

String vemail= rs.getString("vemail");

String vaddr = rs.getString("address");

String cname = rs.getString("company");

String mob = rs.getString("vmob");

VendorBean vendor = new VendorBean(vid,vname,mob,vemail,vaddr,cname,pass);

session.setAttribute("vendordata", vendor);

RequestDispatcher rd = request.getRequestDispatcher("vendorHome.jsp");

rd.forward(request, response);

}

else{

ps = conn.prepareStatement("select \* from vendor where vemail=? and password=?");

ps.setString(1, uname);

ps.setString(2, pword);

rs1=ps.executeQuery();

if(rs1.next()){

HttpSession session = request.getSession();

session.setAttribute("user","user");

session.setAttribute("username", uname);

session.setAttribute("password", pword);

String vid = rs1.getString("vid");

String pass = pword;

String vname = rs1.getString("vname");

String vemail= rs1.getString("vemail");

String vaddr = rs1.getString("address");

String cname = rs1.getString("company");

String mob = rs1.getString("vmob");

VendorBean vendor = new VendorBean(vid,vname,mob,vemail,vaddr,cname,pass);

session.setAttribute("vendordata", vendor); //We need the user data whole the session

RequestDispatcher rd = request.getRequestDispatcher("vendorHome.jsp");

rd.forward(request, response);

}

else{

PrintWriter pw = response.getWriter();

RequestDispatcher rd = request.getRequestDispatcher("login.jsp");

rd.include(request, response);

pw.print("<script>document.getElementById('show').innerHTML = 'Invalid Username or Password<br>Please Try Again!'</script>");

}

}

}catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

finally{

DBUtil.closeConnection(ps);

DBUtil.closeConnection(pst);

DBUtil.closeConnection(rs);

DBUtil.closeConnection(rs1);

}

}

}

}