International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

WEB BASED CIVIL ENGINEERING CALCULATOR

A. Sivaji[1], M.C Venkata Subbaiah[2], K. Arun Kumar[3]

[1,2]Annamacharya Institute of Technology & Sciences (Autonomous),
Department of Civil Engineering
Rajampet, Kadapa Dist., AP, (India)
[3]Annamacharya Institute of Technology & Sciences (Autonomous),
Department of Information Technology
Rajampet, Kadapa Dist., AP, (India)

ABSTRACT

A web application is an application that is accessed over a network such as the Internet or an intranet. The term also means a computer software application that is coded in a browser-supported language and reliant on a common web browser to render the application executable. In This Present study, a web based Calculator for some Civil Engineering Calculations are developed. The calculations are based on codal provisions. The relevant data will be stored in a database. For the present work open source software Eclipse, Apache Tomcat server, Oracle database and SQL Developer are used for developing the application in java programming language.

KEYWORDS: Web browser, Eclipse, Apache Tomcat, Oracle database and SQL.

I.INTRODUCTION

We are living in a Golden Age of software development. The Internet has opened up fantastic new opportunities for applications. There is an abundance of powerful, inexpensive personal computers and mobile devices that can access the Internet and run these new applications. And, to fuel their creation and deployment, the Open Source movement has created an unprecedented array of high quality, freely available middleware and tools. It is truly a great time to be a software developer. We are limited only by our imagination and our ability to master the skills demanded by this rich environment.

A web application is an application that is accessed by users over a network such as the Internet or an intranet. The term may also mean a computer software application that is coded in a browser-supported programming language (such as JavaScript, combined with a browser-rendered markup language like HTML) and reliant on a common web browser to render the application executable.

Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client, sometimes called a thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility. Common web applications include webmail, online retail sales, online auctions, wikis and many other functions.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

II. MIX DESIGN

2.1 Indian Standard Method of Concrete Mix Design (IS 10262 – 2009)

This Standard Provides the guidelines for proportioning Concrete mixes as per the requirements using the concrete making materials including other supplementary materials identified for this purpose. The proportioning is carried out to achieve specified age, workability of fresh concrete and durability requirements.This standard is applicable for ordinary and standard concrete grades only.All requirements of IS 456 in so far as they apply, shall be deemed to form part of this standard.

Data for Mix Proportioning:

The following data are required for mix proportioning of a particular grade of concrete:

- Grade designation
- Type of cement
- Maximum nominal size of aggregate
- Minimum cement content
- Maximum water-cement ratio
- Workability
- Exposure conditions as per IS 456
- Method of transporting and placing
- Type of aggregate
- Maximum cement content
- Type of admixture

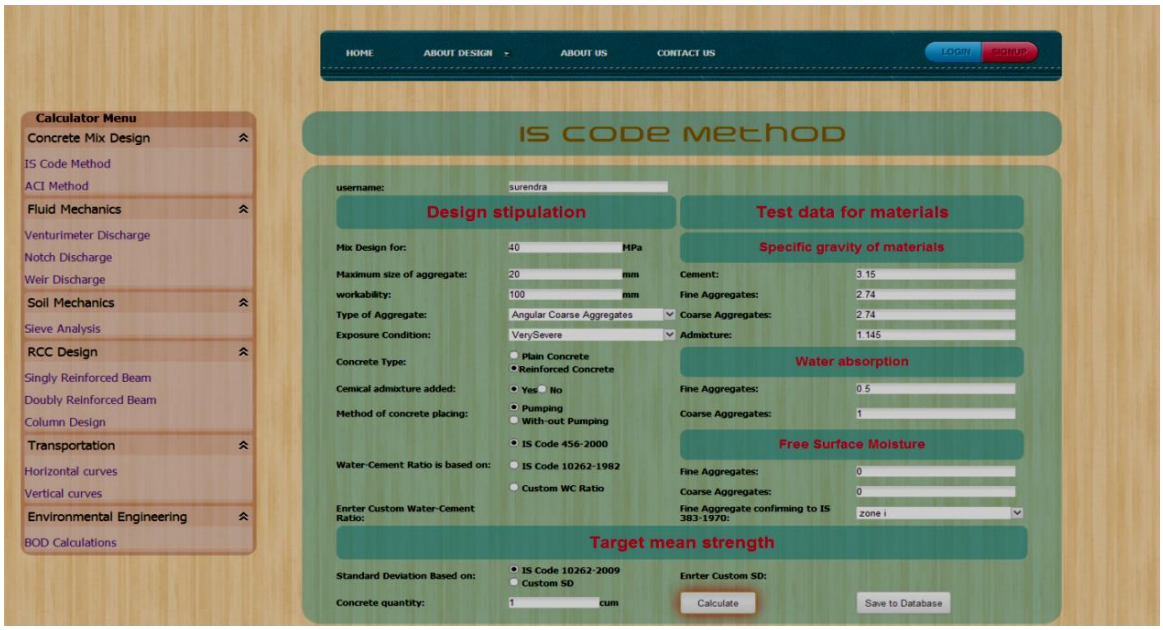

Fig.1 Image of an input jsp file for IS code mix design

Fig.2 Image of an output jsp file for IS code mix design

This concrete mix design based on IS 10262-2009 has been checked manually with the presently developed web application and the results obtained were effectively matched.

2.2 American Concrete Institute Method of Mix Design (ACI211-91)

This Standard Practice describes methods for selecting proportions for hydraulic cement concrete made with and without other cementitious materials and chemical admixtures. This concrete consists of normal and high density aggregates (as distinguished from light weight aggregates) with workability suitable for usual cast-in-place construction (as distinguished from special mixtures for concrete products manufacture). Also include description of methods used for selecting proportions for mass concrete. Hydraulic cements referred to in this Standard Practice are Portland cement (ASTMC150) and blended cement (ASTMC595).The Standard does not include proportioning with condensed silica fume.

The methods provide a first approximation of proportions intended to be checked by trial batches in the laboratory or field and adjusted, as necessary, to produce the desired characteristics of the concrete.

Data for Mix Proportioning:

The following data are required for mix proportioning of a particular grade of concrete:

- Grade designation
- Exposure condition
- Maximum nominal size of aggregate
- Maximum water-cement ratio
- Workability
- Type of admixture
- Fineness modulus of selected F.A.
- Unit weight of dry rodded coarse aggregate
- Sp. Gravity of coarse and fine aggregates in SSD condition
- Absorption characteristics of both coarse and fine aggregates
- Specific gravity of cement.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

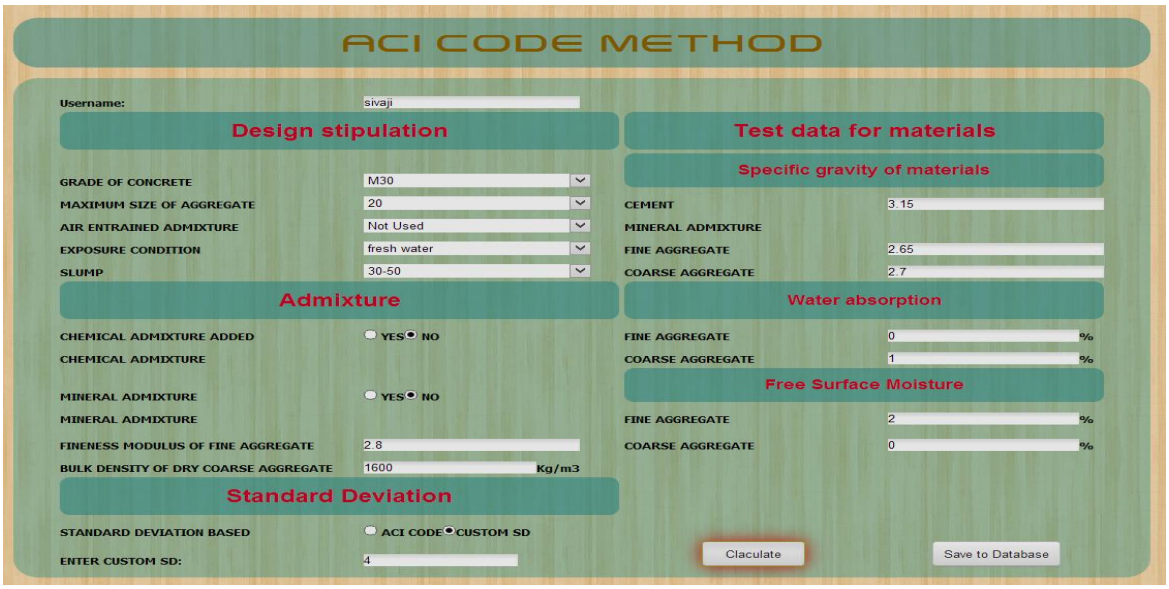

Fig.3 Image of an input jsp for ACI based mix design

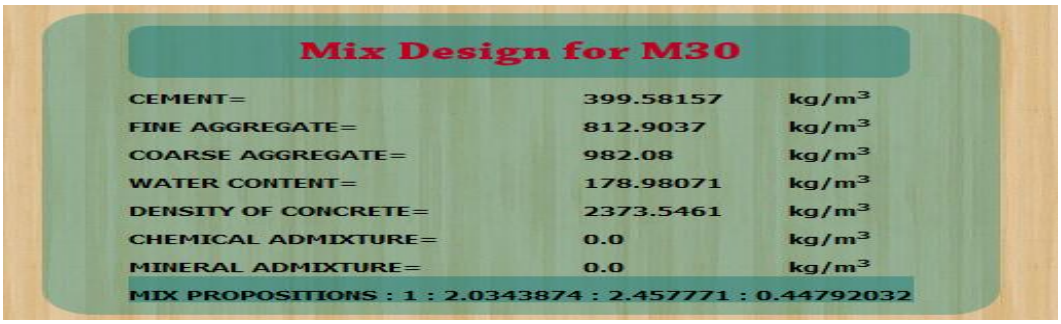

Fig.4 Image of an output jsp for ACI code based mix design

This concrete mix design based on ACI 211-91 has been checked manually with the presently developed web application and the results obtained were effectively matched.

2.3 R.C.C DESIGN:

Design of Singly Reinforced Rectangular Sections

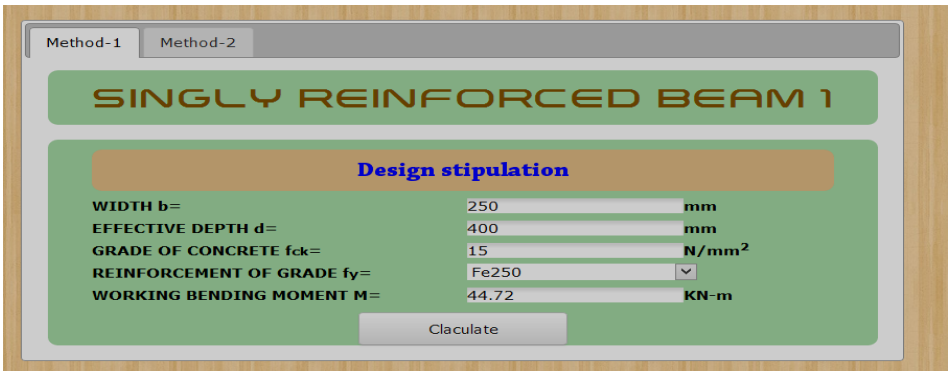

Fig.5 Image of input jsp for design of beams-1

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

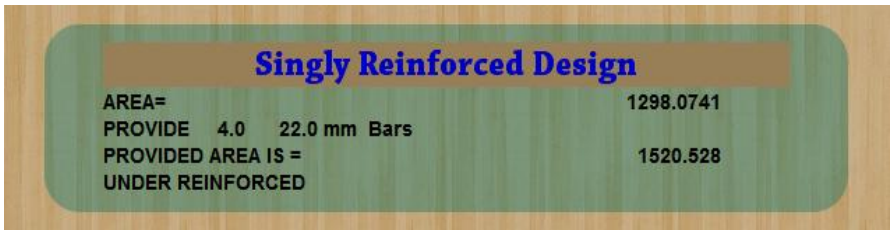

Fig.6 Image of output jsp for Design of beams

Fig.7 Image of input jsp for design of beams

Fig.8 Image of output jsp for design of beams

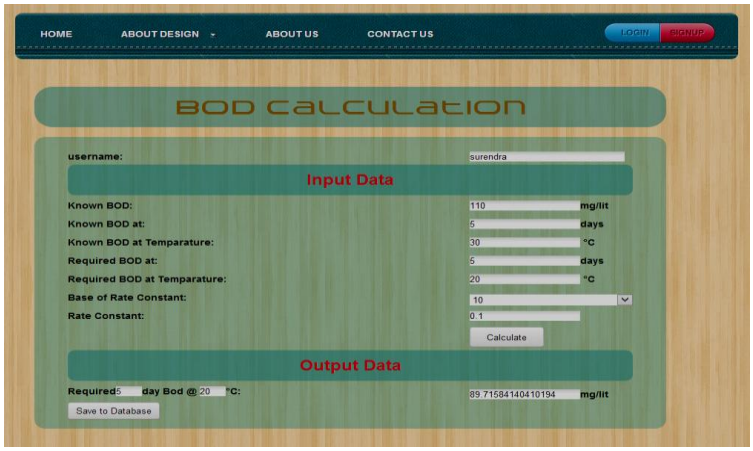

Fig.9 Image of web application for B.O.D. calculation

3. JAVA WEB APPLICATION DEVELOPMENT AND ECLIPSE

We are living in a Golden Age of software development. The Internet has opened up fantastic new opportunities for applications. There is an abundance of powerful, inexpensive personal computers and mobile devices that can access the Internet and run these new applications. And, to fuel their creation and deployment, the Open Source movement has created an unprecedented array of high quality, freely available middleware and tools. It is truly a great time to be a software developer. We are limited only by our imagination and our ability to master the skills demanded by this rich environment.

Java technology and the Java 2 Enterprise Edition (J2EE) have emerged as one of the dominant platforms on which to build Web applications. Numerous Open Source and commercial products support and extend J2EE middleware. And, in the tool arena, Eclipse has emerged as one of the most popular Java integrated development environments (IDE). As the name implies, WTP extends Eclipse into the domain of Web applications. WTP includes both a set of core tools for Web application developers and a set of platform application programming interfaces (API) for tool vendors.

A simple Web application has three basic logical tiers or layers: Presentation, Logic, and Data. More layers can be defined to abstract different parts of the architecture. The physical architecture is an orthogonal concern; all layers can run on the same application server on a single machine, or on three or more application servers on separate machines. J2EE allows us to manage the physical layer independent of the application layers.

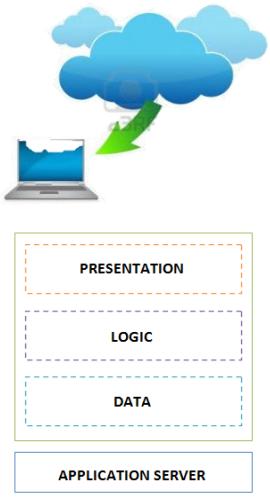

Fig.10 Web Applications and Services

The top layer is for presentation. This user interface layer is often built using HTML. Rich Internet Applications (RIA) and AJAX introduce other client technologies, such as Flash and JavaScript, to this layer. If the user interface does not require anything more than a Web browser to run, it is called a thin client.

The middle layer is where you implement the logic. This would be the layer where is calculations are done using some equations and logics. Typically, this layer is not specific to Web applications. A well-designed system can reuse the model. The bottom layer is where you keep the data in a persistent store. Databases are the most

common choice as they can be kept in a file or a mainframe. Although the Web application is hyperlinked, this is not really integrated. It will form a loosely coupled ecosystem where users act as the binding agent. In a service-oriented system, applications are integrated using services; the users are replaced by other Web applications, and the presentation layer is replaced by the service layer.

3.1 Java Web Applications

Java Web applications use technologies described in the J2EE specification and the more general standards such as HTML, XML, and Web Services.

Layered architectures and client-server designs have been around longer than Java and Web technologies. Probably the most significant architectural contribution of J2EE has been to provide a practical and standardized specification. There are many commercial and Open Source application servers that support this standard.

J2EE provides the standard for the programming and runtime models used for Java Web applications. J2EE has components for client sessions, presentation, application and business logic, and business operations. It has services such as distribution, transaction, and data management to run these components in an enterprise environment.The Web container runs components such as JSPs and servlets. These components are typically used to implement the presentation layer.

3.2 Designing Java Web Applications

1. Accept input
2. Handle application logic
3. Generate output (presentation logic)

This design couples all the layers together, making it difficult to modify or test any particular aspect in isolation. In addition, there are significant issues related to handling these responsibilities. We will describe them next. For servlets used alone, the same issues apply (because we can consider a servlet as a script with some additional embedded text, such as XML or HTML), and mixing code with the text also presents code management and debugging issues.

Accepting Input

When accepting input, a script receives an Http Servlet Request object, which is a minimally parsed representation of the HTTP input stream. HTTP supports three different mechanisms for passing parameters (encoding into the URL, query parameters, and form data), and all of these pass the data as simple strings. Each script must know or determine the parameter-passing mechanism, convert the parameters to appropriate types, and validate them.

Handling Application Logic

Another issue, which affects both input and application logic, is the lack of information hiding when accessing request and session data. The script must retrieve input data from the request by name. Data used in multiple

JSPs must be either stored in a session associated with the user or reread from an external data source in each script requiring the data, as HTTP is a stateless protocol.

Both storage in the session and storage in an external data source are effectively global in scope, and the application accesses the data in a dictionary-like fashion using strings as keys. Normal programming mechanisms for controlling variable access do not apply to this data, and any scripts or server pages that wish to use this data must be aware of the naming conventions. The accessibility to the variables using programming-language mechanisms is not easy task, so modifications become more difficult. If the JSP does not encapsulate these conventions, knowledge of them and details of the HTTP protocol can spread throughout the application, greatly hindering adaptation to new uses. Furthermore, this is a potential source of errors because of both spelling mistakes and reuse of a parameter name for different purposes in different scripts. As the number of JSPs increases, these problems can become overwhelming.

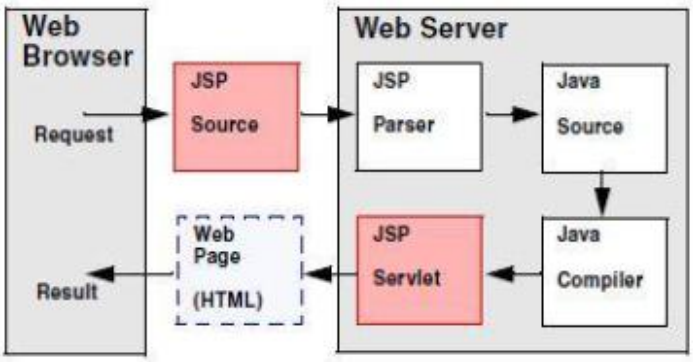

Fig.11 Outline of Control Flow

When using JSPs for application logic, significant amounts of code will be added to the page. Code management techniques are minimal and awkward for code inside JSPs. Debugging code inside the server pages is difficult due to the mix of text with code. Although WTP provides features for code authoring and interactive debugging, for precompiled JSPs debug has to be done inside a complex server-generated code. For these reasons, it is a good idea to minimize the amount of code in JSPs and to keep application logic out of them.

Generating Output

In producing output, simple scripts mix the text, HTML, or XML encoding of the result with the dynamic data. This couples the page's mark-up, its look and feel, with the other layers. Changing the Web site's look or adapting the application to multiple output devices becomes extremely difficult. The latter is becoming increasingly important as the Web expands to include mobile devices such as Internet-connected mobile phones and other embedded devices. JSPs help address this last issue by helping to create and maintain the look and feel of the pages, so presentation logic can be provided in annotations. This is generally considered the most appropriate use for server pages.

IV. DATABASE MANAGEMENT

Database:

A database can be summarily described as a repository for data. This makes clear that building databases is really a continuation of a human activity that has existed since writing began; it can be applied to the result of any bookkeeping or recording activity that occurred long before the advent of the computer era.

Oracle SQL Developer is a graphical version of SQL Plus that gives database developers a convenient way to perform basic tasks. We can browse, create, edit, and delete (drop) database objects; run SQL statements and scripts; edit and debug PL/SQL code; manipulate and export (unload) data; and view and create reports. A connection is a SQL Developer object that specifies the necessary information for connecting to a specific database as a specific user of that database. There should be at least one database connection (existing, created, or imported) to use SQL Developer. Once connected to any target Oracle database schema using standard Oracle database authentication. It can perform operations on objects in the database.

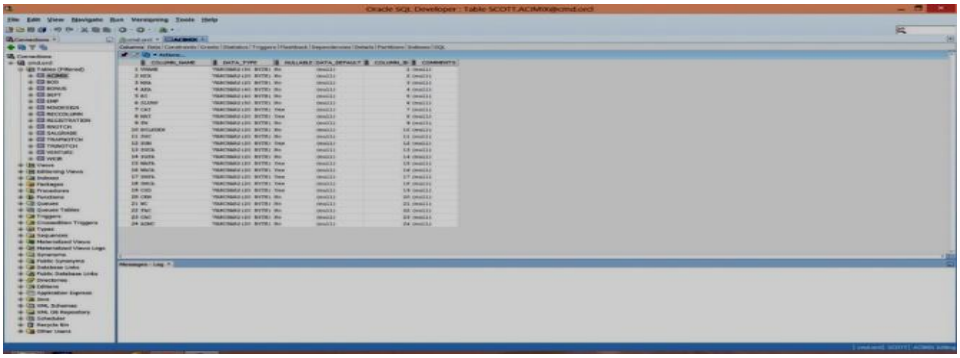

Fig.12 Screen shot of SQL developer

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.

JDBC 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.

Prepared Statement – 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.

Callable Statement – 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.Here in the present application query is to filter the data based on the username and this is available to only those who have registered account. This can be seen in the below figure that how a login and signup screen look like and after login how a filtered result set of the data from a database is displayed on a jsp.

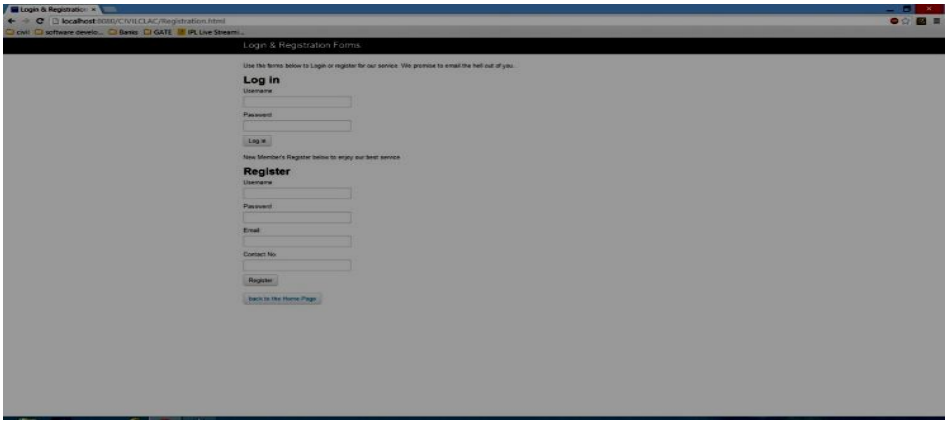

Fig.13 Screenshot of login and registration page

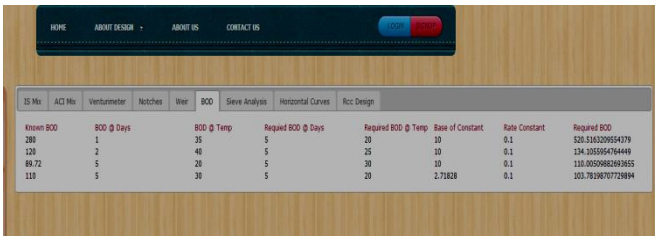

Fig.14 Image of jsp displaying user data

V. CONCLUSION

The programming done gives the results appropriately. It is a new approach to civil and structural engineering software as there is nothing to download, install or update. You can simply open a browser, login and start working.It has the Freedom to work anywhere and at any time As Web application is always available day or night and you can use it on any machine with internet. So, you are not restricted to one machine with one

software license or require multiple licenses or to working at the location with the licensed software or by availability of the software.

Scope for Future works:

There is a plethora of scope for extending this application with certain modifications some of them are mentioned below:

- Deploying web application on the server.
- Inclusion of some more Civil Engineering calculations.
- Improving Database management.

REFERENCES

Standards

[1] *Indian Standard* CONCRETE MIX PROPORTIONING – GUIDELINES *(First Revision)* IS 10262:2009. Bureau of Indian Standards. 2009.

[2] *Indian Standard* PLAIN AND REINFORCED CONCRETE - CODE OF PRACTICE (*Fourth Revision)* IS 456-2000. Bureau of Indian Standards. 2000.

[3] *Indian Standard* SPECIFICATION FOR COARSE AND FINE AGGREGATES FROM NATURAL SOURCES FOR CONCRETE (*Second Revision).* IS 383:1979 (Reaffirmed 1997) Bureau of Indian Standards. 1997.

[4] American standard CONCRETE MIX PROPORTIONING – GUIDELINES (second Revision) ACI 211-1991.

[5] Standard Specification AIR-ENTRAINING ADMIXTURES FOR CONCRETE - GUIDELINES (third Revision) ASTM C 260.

[6] Standard Specification for CHEMICAL ADMIXTURES – GUIDELINES (first Revision) ASTM C1017C1017M.

Books

[7] *Concrete Technology Theory and Practice*M.S. Shetty. S. Chand & Company Ltd. 2005. ISBN: 81-219-0003-4.

[8] *Concrete-Technology*GhanshamBansal. Indian Railway Institute of Civil Engineering, Pune. 2007.

[9] Highway Engineering, Dr. S. K. Khanna, Dr. C. E. G. Justo, Nem Chand & Bros.,Roorkee. 2010. ISBN: 978-81-85240-63-3.

[10] *A Text Book of Fluid mechanics and Hydraulic Machines*Dr. R. K. Bansal, Laxmi Publications (P) LTD,ISBN: 81-7008-311-7.

[11] *Sewage Disposal and Air pollution Engineering* S. K. Garg, Khanna Publishers,

[12] ISBN: 978-81-7409-230-4.

[13] *Reinforced Concrete Design* S. UnnikrishnaPillai, DevdasMenon ,Tata McGraw-Hill Education, ISBN: 9780070141100.

[14] *Soil Mechanics and Foundation Engineering*Dr. K. R. Arora, Standard Publishers and Distributers, ISBN: 81-8014-112-8.

[15] *Eclipse Web Tools Platform*Naci Dai, Lawrence Mandel, Arthur Ryman. Pearson Education, Inc. 2007. ISBN: 0-321-39685-5.

[16] *Eclipse Modelling FrameworkA Developer's Guide* Frank Budinsky, David Steinberg, Ed Merks, Raymond Ellersick, and Timothy J. Gross. Addison- Wesley. 2004. ISBN: 0-13- 142542-0.

[17] *JSP™: Java Server Pages™* Barry Burd. M&T Books, 2001, ISBN: 0-7645-3535-8.

[18] *Oracle® Database SQL Reference Language* Diana Lorentz, B28286-06

[19] *Oracle® DatabaseConcepts11g Release 2 (11.2)* Lance Ashdown, Tom Kyte, E25789-01

Articles

[20] "Apache Derby Fortune Server Tutorial." Jean Anderson. April 21, 2006. Apache Software Foundation. http://db.apache.org/derby/papers/fortune_tut.html