**Table of Contents**

Acknowledgement i

Abstract ii

**1. Introduction..................................................................................................................2**

1.1Purpose...........................................................................................................................2

1.2Scope..............................................................................................................................2

1.3Technologies Used.........................................................................................................4

1.4Overview........................................................................................................................5

**2. Overall Description.......................................................................................................6**

2.1ProductFeatures............................................................................................................6

2.2Operating Environment................................................................................................6

2.3Design and Implementation Constraints......................................................................7

**3. Functional Requirement...............................................................................................8**

**4. External Requirement..................................................................................................9**

4.1User Interfaces.............................................................................................................9

4.2Hardware Requirement...............................................................................................9

4.3Software Requirement…..............................................................................................9

**5. Others Non-Functional Requirements..........................................................................11**

5.1 Usability:……….………...................................................................................................11

5.2 Reliability……….…........................................................................................................11

5.3 Performance…..…….....................................................................................................12

5.4 User Interface...........................................................................................................12

5.5 Operation……………………………………………………………………………………………………………..12

5.6 Supportability……………………………………………………………………………………………………….12

5.7 Implementation………………………………………………………………………..............................13

**6.Design Details............................................................................................................13**

6.1 Class diagram.........................................................................................................13

6.2 Use Case Diagram..................................................................................................15

6.3 Sequence Diagram.................................................................................................17

6.4Activity Diagram.....................................................................................................19

6.5E –R Diagram...........................................................................................................21

**7.References................................................................................................................34**

**CHAPTER 1**

**Introduction**

We have three roles in this system administrator, critics and user. Administrator logs into this system, and can register a critic who belongs to that genre of field. User check in this system and listen to the latest tracks.

Critics rate the tracks, which are made available by the administrator according to the in-built algorithm and newly released tracks. These tracks are then ranked according to these rating and showcased on the system where any user can see the ranking of the song and will know where it stands.

In the same system the user can access to the song i.e. they can play it, get the information of its singer, albums, ranking and much more.

* 1. **Purpose :**
* This Web Application provides chart containing top songs of all the genre.
* It saves time as it allows all type of genre in a single system so user doesn’t need to browse all over the web in search for best music.
* Administrator has privileges to select the kind of content. It includes songs, comments of user and critic to maintain the quality of the application.
* Users can find it easy to access top songs from many from one platform.
  1. **Scope :**

The scope of this project is developing a web-based system. And the developed web-based system manages the activity of “Critics rating Management” and “Online Showcasing top songs”. ”Beatrix” system will manage the database and maintain a list of all songs that have on this site or currently released. System can efficiently perform all the processing, which include recently released songs. The chart thus produce presents the most quality tracks ow the week.

* 1. **Technology Used :**

**Front – End :-**

* HTML
* CSS
* JavaScript
* Bootstrap

**Back – End :-**

* JavaServer Pages
* MySQL

**HTML :-**

“**HyperText Markup Language**”, is a Universal language which allows an individual using special code to create web pages to be viewed on the internet.

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

 HTML describes the structure of Web pages using markup

 HTML elements are the building blocks of HTML pages

 HTML elements are represented by tags

 HTML tags label pieces of content such as "heading", "paragraph", "table", and so.

HTML also defines special **elements** for defining text with a special **meaning**.

It provides a means to create structured documents including headings, pictures, objects, lists, links, and others items and can be used to create interactive pages. It can include or can load scripts in languages such as JavaScript which affects the behavior of HTML processors like Web Browsers.

* **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus the link available on a webpage are called Hypertext.
* As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Everyone has a different preference for which tool works best for them. Keep in mind that typically the less HTML the tool requires you to know, the worse the output of the HTML. In other words, you can always do it better by hand if you take the time to learn a little HTML.

**HTML5** is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a mark up language.

**HTML5** is designed, as much as possible, to be **backward compatible** with existing web browsers. New features build on existing features and allow you to provide fallback content for older browsers.

It is suggested to detect support for individual HTML5 features using a few lines of JavaScript.

The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.

HTML 5 does not have the same syntax rules as XHTML where we needed lower case tag names, quoting our attributes, An attribute had to have a value and to close all empty elements.

HTML5 Advantages :

* Cleaner markup/ improved code
* Elegant forms
* Consistency
* Supports rich media elements
* Offline application cache

**CSS :-**

“**Cascading Style Sheets**” fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS was invited by **Håkon Wium Lie** on October 10, 1994.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of others effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

* **CSS saves time** − You can write CSS once and then reuse same heet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style heets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So it’s a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
* **Offline Browsing** − CSS can store web applications locally with the help of an offline catche. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
* **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.CSS3 is a latest standard of css earlier versions(CSS2).The main difference between css2 and css3 is follows

* Media Queries
* Namespaces
* Selectors Level 3
* Color

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –

* Selector
* Property
* Value

CSS3 is collaboration of CSS2 specifications and new specifications, we can called this collaboration is **module**. Some of the modules are shown below

* Selectors
* Box Model
* Backgrounds
* Image Values and Replaced Content
* Text Effects
* 2D Transformations
* 3D Transformations
* Animations
* Multiple Column Layout
* User Interface

**MySQL :**-

MySQL the most popular open source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.

**MySQL database are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

**MySQL software is open source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

## What is Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Others kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

## MySQL Database:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## Installing MySQL on Windows:

Default installation on any version of Windows is now much easier than it used to be, as MySQL now comes neatly packaged with an installer. Simply download the installer package, unzip it anywhere, and run setup.exe.

Default installer setup.exe will walk you through the trivial process and by default will install everything under C:\mysql.

Test the server by firing it up from the command prompt the first time. Go to the location of the mysqld server which is probably C:\mysql\bin, and type:

mysqld.exe --console

**NOTE:** If you are on NT, then you will have to use mysqld-nt.exe instead of mysqld.exe

If all went well, you will see some messages about startup and InnoDB. If not, you may have a permissions issue. Make sure that the directory that holds your data is accessible to whatever user (probably mysql) the database processes run under.

MySQL will not add itself to the start menu, and there is no particularly nice GUI way to stop the server either. Therefore, if you tend to start the server by double clicking the mysqld executable, you should remember to halt the process by hand by using mysqladmin, Task List, Task Manager, or others Windows-specific means.

**JavaScript :-**

JavaScript is a programming language that can be included on web pages to make them more interactive. You can use it to check or modify the contents of forms, change images, open new windows and write dynamic page content. You can even use it with CSS to make DHTML(Dynamic Hyper Text Markup Language). This allows you to make parts of your web pages appear or disappear or move around on the page. JavaScripts only execute on the page’s that are on your browser at any set time. When the employee stops viewing that page, any scripts that were running on it are immediately stopped. The only exceptions are cookies or various client side storage APIs, which can be used by many pages to store and pass information between them, even after the pages have been closed.

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

## What is JavaScript ?

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and others web browsers.

## Client-side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and others actions that the user initiates explicitly or implicitly.

## Advantages of JavaScript

The merits of using JavaScript are −

* **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

## Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multithreading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

**JavaServer Pages :**

JavaServer Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

JavaServer Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc.

## Why Use JSP?

JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI.

* Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files.
* JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
* JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc.
* JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

## Advantages of JSP :

Following is the list of others advantages of using JSP over others technologies:

* **vs. Active Server Pages (ASP):** The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or others MS specific language, so it is more powerful and easier to use. Second, it is portable to others operating systems and non-Microsoft Web servers.
* **vs. Pure Servlets:** It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.
* **vs. Server-Side Includes (SSI):** SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.
* **vs. JavaScript:** JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.
* **vs. Static HTML:** Regular HTML, of course, cannot contain dynamic information.

A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.

The following are the paths followed by a JSP

* Compilation
* Initialization
* Execution
* Cleanup

The four major phases of JSP life cycle are very similar to Servlet Life Cycle and they are as follows:



## JSP Compilation :

When a browser asks for a JSP, the JSP engine first checks to see whethis it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

* Parsing the JSP.
* Turning the JSP into a servlet.
* Compiling the servlet.

## JSP Execution :

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.

Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the **\_jspService()** method in the JSP.

The \_jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods ie. GET, POST, DELETE etc.

## JSP Cleanup :

The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The **jspDestroy()** method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.

* 1. **Overview :-**
* The online testing system, which starts automatically and stops after a particular stipulated time.
* Auto generation of emails as soon as the professor assigns projects to the student groups, intimating them that they have been assigned project and need to report to the professor.
* Validations like the student doesn’t give the test twice etc.

**CHAPTER 2**

**Overall Description**

**2.1 Product Features :**

* Administrator can login to the system and decide the critic panel and also the kind of tracks considered for the ratings.
* Administrator can create user account for a critic and provide songs to critic .
* Critic can login and give the ratings to the tracks made available to them.
* Users can check all the new songs, listen them and see the rankings.

**2.2 Operating Environment:-**

* Operating System (Window 7)
* web browser
* database management system(MySQL)
* java database connectivity

2.**3 Design and Implementation Constraints :-**

**INPUT SCREENS**

1. THE LOGIN PAGE:

The login page should have fields such as Login ID and Password.

1. THE REGISTRATION FORM:

The Registration form should have fields such as Login id, Password, Confirm Password, Name, Age, Group mail-id etc. for each group of critics.

1. RATING THE SONG:

The critic is give the ratings to the songs

**OUTPUT SCREENS**::

1. THE VIEW PROFILE MODULE:

This module should provide a view of the registered critics. The login id, user name id, Name, e-mail id, etc.

2) THE EDIT PROFILE MODULE:

The Edit profile module should have fields such as Login id, Password, Confirm Password, User name. One can make changes to the data entered.

1. THE EDIT PASWORD MODULE:

The user can change its password through this module.

1. THE MUSIC CHART

This module provide the rankings of the songs with additional functions i.e. play/pause’

code.

**CHAPTER 3**

**Functional Requirement**

**3.1 Performance :-**

The Beatrix management system operates its function in small amount of time which is less then seconds and can be accessed by many users at a time.

**3.2 Reliability:-**

The Beatrix is available based on the user needs, can work properly, and do transactions efficiently including safe data management. Here the Administrator control over the system by examining critics rating and daily released songs. Any user can’t use the system but the guest user can see general properties of the Beatrix. As result data is protected and controlled by only the Administrator

**3.3 Maintainability:-**

The Beatrix web application is easily maintainable and this can be done by administrator who is handling the web application and adding all the required fields and components such as Songs, Cover Images etc by easily updating it in the Database.

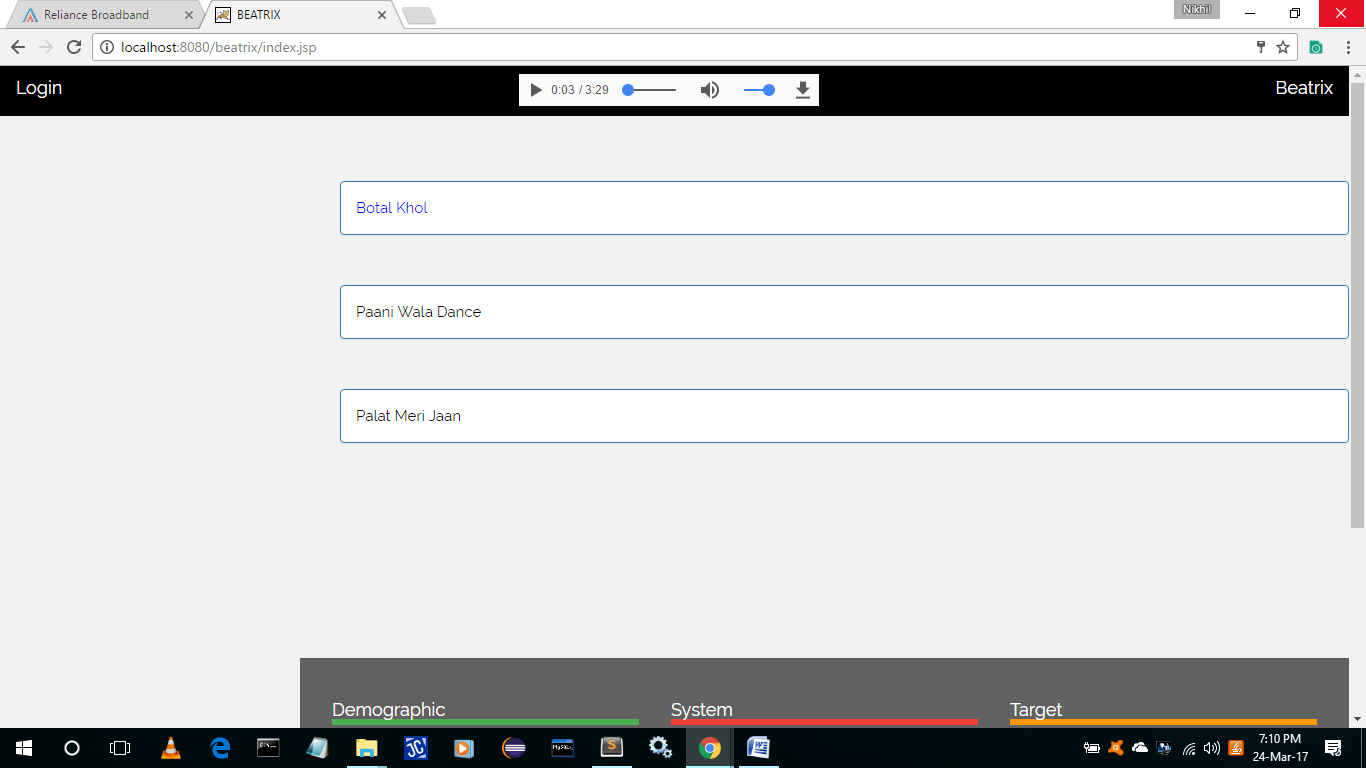
**3.4 Supportability:-**

The Beatrix operates in any version of windows operating system. Such as windows xp, windows 2003, windows 7, windows 8 and others related versions. The system can be easily maintained by the Administrator of this system by using the prepared documents of the system for easy maintenance. Others ways it is maintained by the system developers for corrective and others heavy problems.

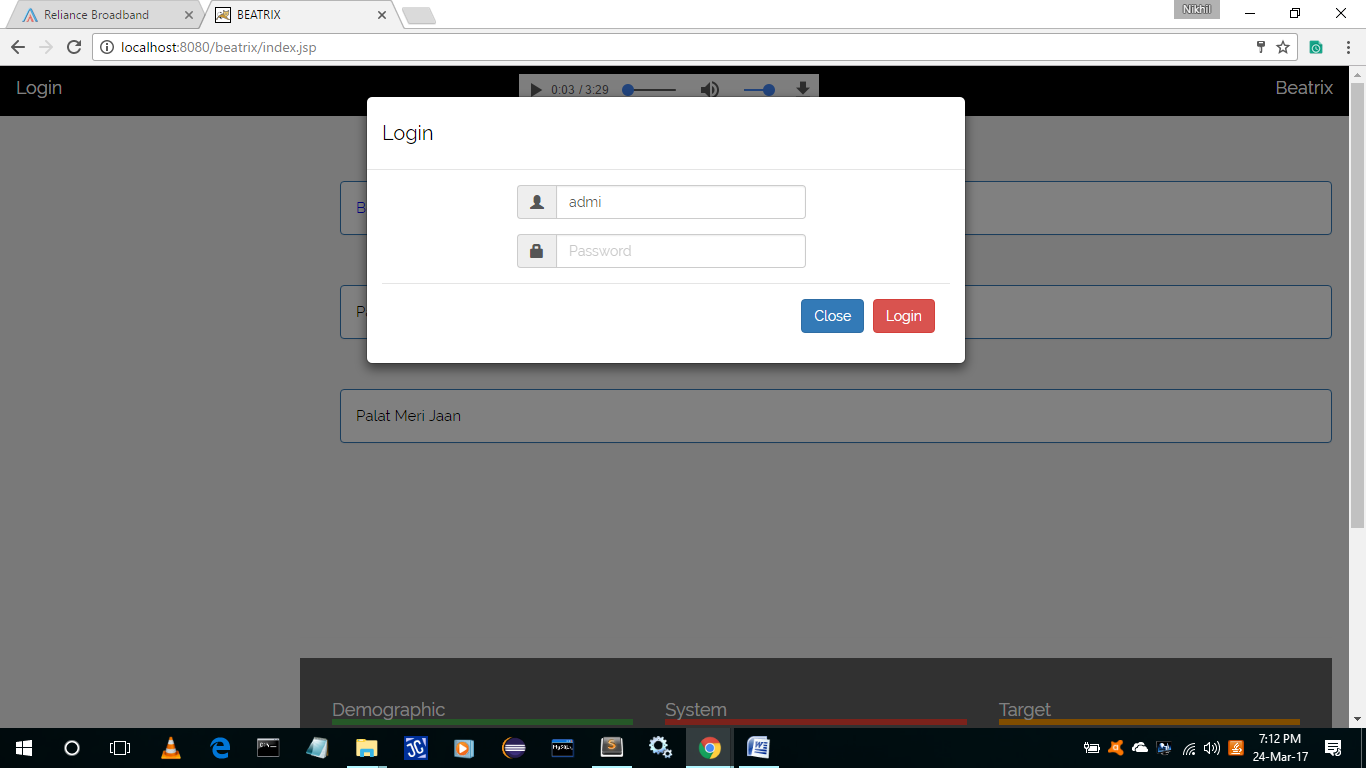
**CHAPTER 4**

**External Requirement**

**4.1 User Interfaces:-**

****

**Figure 4.1.1(**Home page Displaying Top Songs**)**

****

**Figure 4.1.2(**Login Page for Critics and Admin**)**

4**.2 Hardwre interfaces :**

* PC with 2 GB hard-disk and 256 MB RAM
* University network infrastructure (wired and wireless)
* Mac, Unix and Windows employee computers
* The environment that will host the university-wide databases
* Web server computers and related hardware support (back-up tapes, redundant drives, etc.)

**4.3 Software interfaces :**

* Windows 95/98/XP with MS-office
* MS-SQL server, MS-Access
* Network software and protocols in order for systems to communicate: TCP/IP, HTTP, HTTPS.
* To use web server, application server and database software in development.

**CHAPTER 5**

**Non-Functional Requirements**

Beatrix System is able to operate in the following characteristics :

**5.1 Usability** :-

Any familiar in using windows operation can operate the system since it have user friendly user interface. Which have the instruction menu’s how to use it which self friendly user interface. Which self direction application then can be used the system without ambiguity.

**5.2 Reliability :-**

The Beatrix is available based on the user needs, can work properly, and do transactions efficiently including safe data management. Here the Administrator control over the system by examining critics rating and daily released songs. Any user can’t use the system but the guest user can see general properties of the Beatrix. As result data is protected and controlled by only the Administrator.

**5.3 Performance :-**

The Beatrix management system operates its function in small amount of time which is less then seconds and can be accessed by many users at a time.

**5.4 user interface :-**

The user interface is friendly which is easy to use. And having attractive frames structure which is prepared in assumption with others related system. Also the user can change him/his user favourite interface that is available in the system.

**5.5 operation :-**

The Beatrix is operated and controlled by the Administrator for safe work.

**5.6 Supportability :-**

The Beatrix operates in any version of windows operating system. Such as windows xp, windows 2003, windows 7, windows 8 and others related versions. The system can be easily maintained by the Administrator of this system by using the prepared documents of the system for easy maintenance. Others ways it is maintained by the system developers for corrective and others heavy problems.

**5.7 Implementation :-**

The System is implemented in Intel(R) Core(TM) i3 processor with 2 GB RAM , 32 bit computer . And it is implemented through testing on both Black and White testing. The language we use implement the system is HTML, CSS, JavaScript, MYSQL, JavaServer Pages.

**CHAPTER 6**

**DESIGN DETAILS**

**6.1 Class Diagram:-**

**6.2 Usecase Diagram:-**

**6.3 Sequence Diagram:-**

**6.4 Activity Diagram**