***S***

***YNOPSIS***

**THE TITLE OF THE PROJECT**

***MUSIC STORE MANAGEMENT***



**INTRODUCTION**

In today’s scenario computer is an important part day-to-day life. Every individual right

from a college student to a business tycoon utilizes the computer to browse the internet to

send email, chat, buy products via net etc.

This is inspired and encourage us to develop application software for every individual.

Our project entitle **“MUSIC STORE MANAGEMENT”** is designed that the Client’s requirement deals with opening an MUSIC STORE Management that will enable the client to mobilize its business through buying the different music related items in the website. It will enable a customer to have access to online shopping that is the site should be dynamic and more customer centric business procedure thereby maximizing the profit of the organization.

**“MUSIC STORE MANAGEMENT”** acts as a virtual showcase for a electronic music shop giving easy access to customers through login procedure, to interact with the **DATABASE**. The proposed solution will be developed using JSP for building the interfaces/Presentation Layer, EJB for business components/Logical Classes and Oracle8i as the database.

This is portal based automation project, which provides communication between the various users for music items such as music CD’s/Cassettes. User can purchase their needed music item anywhere, anytime just clicking their mouse. This project increases the interest of purchasing. User can easily finished their shopping through this shopping cart.

Screen design/Graphical User Interface:-

Graphical User Interface (GUI) that is straightforward and easy to navigate has been designed. This GUI provide various screens with appropriate incorporate icons, hyperlinks etc. to facilitate screen navigation and data entry. The user can easily add items to their cart, and they can easily remove items form the cart if they needed. The user has the ability to return to home page from any location within the application.

The following GUI form for user interaction can be conferred in the **“MUSIC STORE MANAGEMENT”** system

The **“MUSIC STORE MANAGEMENT”** is divided into three modules:

* + **Visitor module**
  + **User module**
  + **Administration module**

**Visitor:-**

If the client is a new user of the system then he performs the following tasks:

* New user can visit the home page of the online music site
* Search for a particular music item
* Cast his vote for a particular song
* View new releases
* For purchasing items, first the visitor has to click the register option in the client login area.
* The user must fill the registration form. The data are stored in to the database. It is not possible to leave any required data from the registration form.
* After that the visitor becomes a user and then he can enter into the shopping cart.
* View the Help page

Once the system authorizes the new client, then he can perform all the functions of the registered user.

**User: -**

If a registered user uses the following application then she/he has to perform the following tasks:

* First he/she has to login into the user interface which will help him/her to avail the services of the shopping cart.
* For the client to log in to the system it has to provide its username and password for authorization purpose.
* If the username and the password match with the entry in the database then the client enters the client shopping area. The client shopping area has three options to be utilized.
* The first option gives the client to search the list of music CD’s/Cassettes offered by the shop where the client can select the required one.
* The second option gives the client to add items to shopping cart.
* The third option gives the client to give feedback.
* The third option gives the client to logout from the site.

**Administrator:-**

The administrator has the following privileges:

* Delete a particular user from the database
* Add particular item to the inventory list
* Modify the details regarding a particular music item
* View the report, which includes sales reports for a specified day, previous week, and previous month.
* View messages and feedback from the customers

**FORMS:-**

We have different types of forms available for different module. These forms are listed and explained below.

**Visitor Module:**

This module has the following pages:

* The Home Page
* The Chart Toppers Page
* The Search Page
* The Vote Page
* The New Release Page
* The Register Page
* The Login Page

**TOOLS/ PLATFORM USED**

**HARDWARE SPECIFICATION**

Processor : Pentium-III

Memory : 128MB

Hard Disk : 20GB

Floppy Drive : 1.44MB

Monitor : 14’’ or above

Mouse : Standard Mouse

Key Board : 104 keys

**SOFTWARE SPECIFICATION**

Operating System  **:**  WINDOWS 2000 Server

Programming Language  **:** JAVA

Server Technologies  **:**  J2EE (JSP, SERVLETS, JDBC)

Application Server  **:** Jakarta Tomcat 4.1

Database Support  **:** ORACLE 8i

**ANALYSIS**

**DATA FLOW DIAGRAM**

**Data Flow:**

An arrow represents a data flow; it represents the path over which data travels in the system. A data flow can move between processes, flow into or out of data stores, to and from external entities.

**Bubbles (Process):**

A circle or bubble represents that transforms data from once form to another by performing some tasks with the data.

**Data store:**

A data store is a place where data is held temporarily from one transaction to the next or is stored permanently.

**External Entity**:

Which defines a source or destination of system data also called an external entity. Based on the working process of the proposed system Data Flow Diagram (DFD), is a model, which gives the insight into the information domain and functional domain at the same time can be drawn using OMT symbols. DFD is refined into different levels. The more refined DFD is more details of the system are incorporated. In the process of creating a DFD, we decompose the system into different functional subsystems. The DFD refinement results in a corresponding refinement of data.

The DFD of the “MUSIC STORE MANAGEMENT”

Each break-up has been numbered as per the rule of DFD. Here we attempted to incorporate all the details of the system and still it requires further improvement since the entire system is under study.

0 LEVEL D.F.D.:-

This is the context level D.F.D. of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The purposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system.

Context Level Diagram

Valid

communication

Services

User

User

**0 LEVEL D.F.D. FOR “MUSIC STORE MANAGEMENT”**

1ST LEVEL D.F.D.:-

This D.F.D. shows all the processes together with all the data stores (tables). It shows the true data flow i.e. how data is actually flowing in the system. Data is coming from which table and going into which table is clearly shown by this DFD. This DFD is the main reference for the development of the system. After understanding the whole system, the application developer will fall back upon this DFD during the Development phase.

Valid

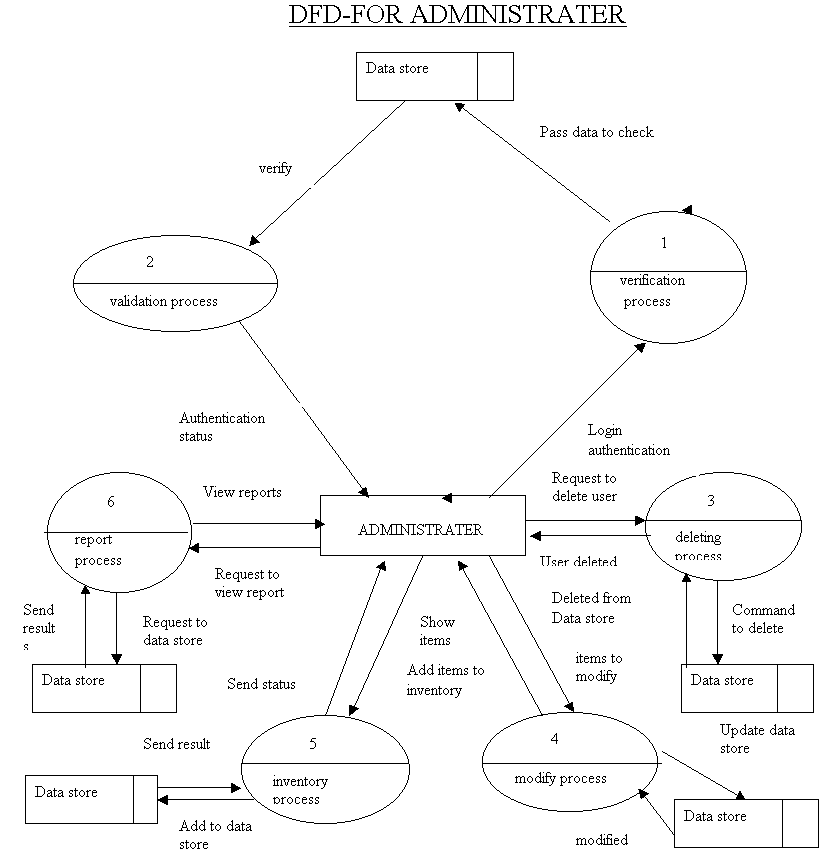
Login

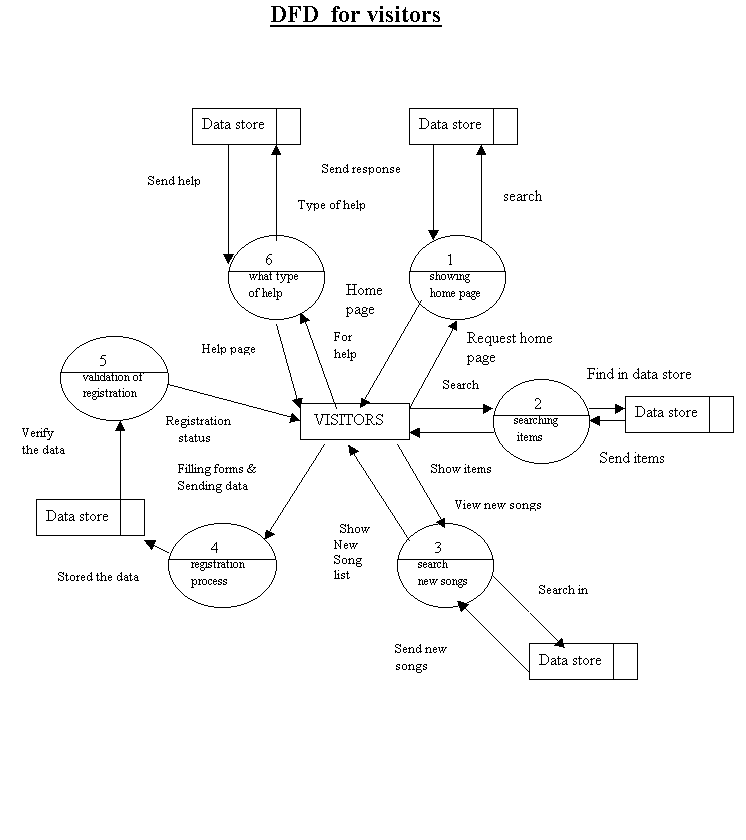
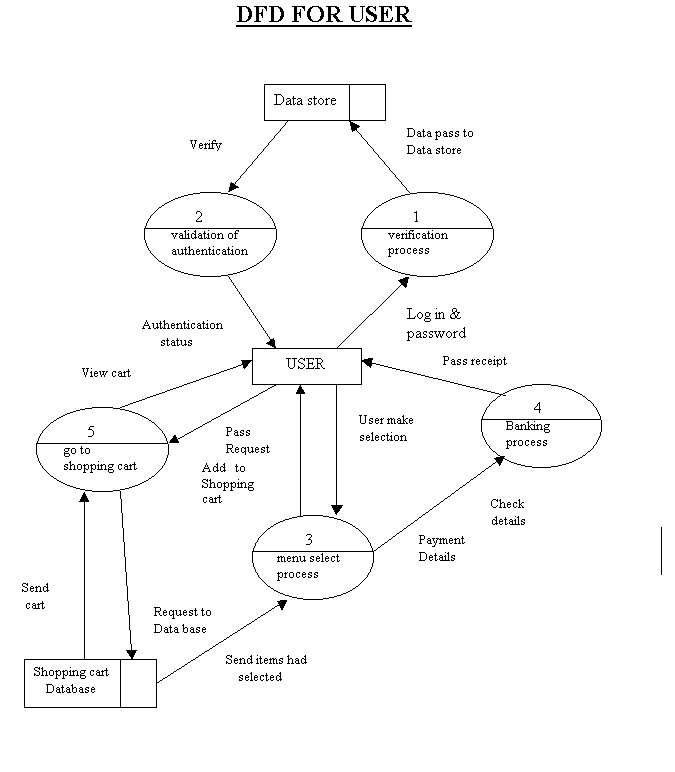
Valid

User

Invalid

Valid

******



**ER-Diagram**

Enter

Login

Information

Feed back

Master

Reply

User information

Registration

Purchases

Reply info

Sales master

Vote

Vote info

Item master

Sale info

[**Database Design**](http://databasesearch.blogspot.com/2010/04/database-fundamentals.html)

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

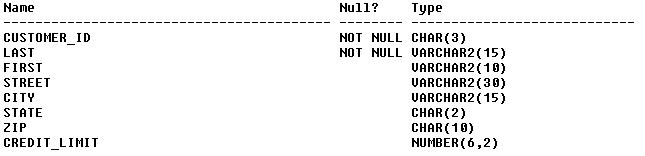
The objectives of the file design are to provide effective auxiliary storage and to contribute to the overall the efficiency of the computer program component of the system. In concepts of database design, there are two types of data – physical data and logical data.

Physical data is that which is written on those pieces of paper. Logical data are those, which are calculated based on some of the retrieved data in a certain sequence in summary form. In a computer-based data processing system, separation of physical and logical data provides the same advantages.

**The Following are Basic Facts about databases.**

|  |  |
| --- | --- |
| **Database Terms** | |
| * Attribute * Cardinality * Data Dictionary * DBMS Engine * Design Tools * Attribute's Domain * Entity * Entity Class * Father of Relational Databases * Foreign Key | * Hierarchy of Data Elements * Meta Data * Overhead Data * Primary Key * Relation * Relational Database * Runtime Subsystem * Schema * Transactions * User Data |

**ATTRIBUTE**  
An attribute is another word for field. In spreadsheet language it would be a cell. It is a place in a database table to store one piece of data of a given type. For example an attribute designated to hold a last\_name, could hold "Smith", but should not hold "Amy Smith".  
  
**CARDINALITY**  
Cardinality is a way to express minimum value and maximum value which are governed by the business rules. Cardinality refers to the required number of instances an entity must have in order to make the in a relationship in order for it to be valid. Minimum cardinality then for a one to many relationship would be one. Minimum cardinality for a basketball team would be 5, or you would be forced to forfeit the game. Maximum cardinality is the maximum number of entities which can occur in a relationship in order for it to be valid. In a one to one relationship the maximum cardinality would also be one. For a baseball team, during the normal season, the maximum cardinality would be 25 active players on the roster.   
  
**THE DATA DICTIONARY**

A database is self describing. By this we mean it documents itself through table structure outputs. One of the components of the data dictionary is the table data type layout.   
   
You can easily see how the fields are defined. The data types, lengths of the fields, and if they can be null or not. This is just one example of the data dictionary information provided by a DBMS.   
  
**THE DBMS ENGINE**

This is a component of the DBMS (Database Management System) which is the intermediary between the design tools and run-time sub-systems and the data. The DBMS engine receives requests form the other two components, which is presented in column and row format, translates them into commands which are passed to the operating system in order to provide read and write functions to the disk.

**THE DESIGN TOOLS**

One of the three components of a DBMS. This subsystem provides the tools to assist users and programmers in creating and modifying components of the database. such components are, Tables, Queries, Reports, and User Forms. Many DBMS products provide a programming environment to create databases which perform very complex tasks.

**Attribute's DOMAIN**

The domain of an attribute is the range of data it can contain. This is not to say the attribute can contain the entire range at one time. An attributes contents must be atomic, meaning they must be of a single bit of information about the theme of the record. For example an attribute named "JOB\_TITLE", from the EMPLOYEES table, could contain values from "Machine Operator", "Driver", "Foreman","Shift Manager", all the way up to "President". It can only hold one of these per record at a time. An attribute designated for "JOB\_TITLE" cannot hold any other type data, such as Salary, or Date\_Of\_Hire. Can you imagine having to look for the Date\_Of\_Hire somewhere in a table, but having no specified place? You might as well be searching text files again.  
  
**ENTITY**  
A entity is something that someone wants to track. An employee for example. It is basically the subject for a table. You gather data about the employee, you run queries to find out information about them, track their time, vacation, sick days etc... Therefore, an entity is very much the same as a record in a table.relational databases was E.F. Codd, who worked for IBM at that time. He published a paper titled "A Relational Model of Data for Large Shared Databanks" in June of 1970.   
  
**ENTITY CLASS**

A entity class is a collection of entities, as defined by their structure. There are usually many entities in an entity class, all of the same structure and type. In my mind, an entity class is the table which contains the entities.

**Father of Relational Databases.**

**FOREIGN KEY**

A foreign key is the same data field and type which is linked to a primary key in a corresponding table. For example in a transaction table The Customer\_ID would be the Foreign Key field. The Foreign Key is used to look up the Customer\_ID in the customer table where the Customer\_ID is the primary key.

**The Hierarchy of Data Elements in:**

* A file processing system
  1. Bits->Bytes or Characters->Fields->Records->Files
* A Database system
  1. Bits->Bytes or Characters->Fields->Records->Files+Metadata + Indexes + App MetaData.

**META DATA**

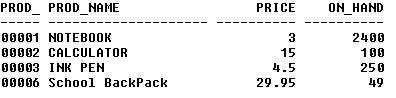
Meta data is the data about data. In the above example concerning the DATA DICTIONARY. Is an example of meta data. It is the self describing part of a database. Information such as the table names, user names, data types, and field sizes are all meta data, describing the database.  
  
**OVERHEAD DATA**

Overhead data is that which the system uses for itself. Indexes for example are overhead data. This is because the system uses indexes to speed searches, and to aid in joins. The overhead part comes in that this data also consumes processing time, and resources. Each time you update a table, the index must also be updated, which takes a bit of processing time but it also speeds up the search capability. You have to decide if the price in resources is worth the benefit of speed in creating and maintaining an index.

**PRIMARY KEY**

The primary key is that field, or fields, which by itself, or together uniquely identify each row in a table. The Primary Key is usually indexed, in some systems that is required. The primary key is normally the field or combined fields by which joins are linked. All data within each row or record should be dependent on the  entirety of the primary key. Primary Keys are used to normalize data tables.

**A RELATION**

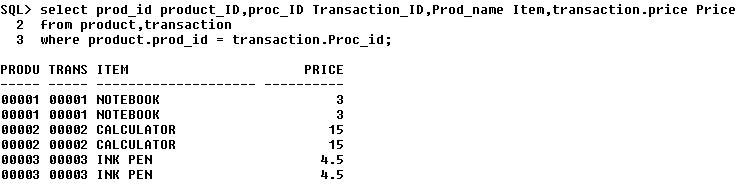
This is a table which, as one of its attributes has a unique identifier for each of it's records, also known as a primary key. In most cases, the primary key is indexed to enhance performance of the system by speeding the lookup capabilities of the DBMS.  
  
In the above example you can see the Product\_ID is unique for each item. This entire table, with the unique field is called a relation.

**A RELATIONAL DATABASE**

Relational database we entails some forms of data relationship. It gets its name for it's relation to other tables within the database. A relational database is set up so that the a key is presented in two or more tables. In one table it will be the primary key, however, in the other table it will be the foreign key. Where the primary key matches the foreign key is where the relationship occurs. You may have a one to one relation where only one of each key members can be present in each table. You may also have a one to many relationship, where only one member can exist in one table but many occurrences can be present in the other table. An finally you can have a many to many relationship, where many occurrences can be present in both tables. Below is an example of a one to many relationship. There is only one occurrence of the Product\_id in the products table, But many occurrences can exist in the transaction table.layout of the tables, attribute types and sizes, which fields are indexed, the relationships, domains, and business rules concerning a database. It is the design from which the database as well as its application programs were built. In a nutshell, the schema encompasses everything about the database.entire function. For example a sales transaction. You need the following.

**THE RUNTIME SUBSYSTEM**

This subsystem processes the application components that are developed using the design tools. For example Access has a runtime component that links data to forms, and reports. This is just part of the DBMS. The user or the developer need be concerned with how it works. When a given form is opened the runtime subsystem opens the required tables extracts the data and displays it to the user. There is also a component that facilitates the read and write requests for the applications.

**SCHEMA**  
The SCHEMA is the design of the database, and why it was created. The schema is the   
  
  
  
**TRANSACTIONS**  
Transactions are a group of sql statements which work together to perform an

1. A statement to add a record to the transaction table.
2. A statement update the Inventory Table.
3. A statement update the customer table. If necessary.
4. A statement to commit the data.

One of two things MUST happen. All of these statements must work together to accomplish their goal, or none of the statements work. That is the key to transaction processing, all or nothing. Log files are kept by the system to record what has been accomplished so, in the event something goes wrong, we know where to start. This is a way of maintaining the integrity of our data.   
  
**USER DATA**  
User data is just as the name implies. User data is the data which the user enters into the database tables.

**A COMPLETE structure**

**Analysis Report:**

System analysis is the first step towards the software building process. The purpose of system analysis is to understand the system requirements, identify the data, functional and behavioral requirements and building the models of the system for better understanding of the system.

In the process of system analysis one should first understand that, what the present system, what it does, is how it works (i.e. processes). After analyzing these points we become able to identify the problems the present system is facing. Upon evaluating current problems and desired information (input and output to the system), the analyst looks towards one or more solutions. To begin with, the data objects, processing functions, and behavior of the system are defined in detail. After this models, from three different aspects of the system-data, function and behavior. The models created during the system analysis process helps in better understanding of data and control flow, functional processing, operational behavioral and information content.

The proposed system contains the following main processes: -

**PROJECT DESCRIPTION**

**PROJECT MODULES:**

The proposed website will have its main page and will be mainly divided into partially dependent and partially independent modules as

**Visitor Module:**

This module has the following pages:

* The Home Page
* The Chart Toppers Page
* The Search Page
* The Vote Page
* The New Release Page
* The Register Page
* The Login Page

**Home Page:**

The Home page of the music application will be the first page to be displayed when a person visits the music application. The page will display a welcome message and, in addition, it will display various menus to facilitate navigation through the application.

**The Chart Toppers Page:**

The Chart Toppers page will display a list of the most popular songs. The popularity of the song will be decided based on the voting at the Vote Page

**The Search Page:**

The Search Page will enable users to perform a search of songs based on the song categories like song name, singer name or album name.

**The Vote Page:**

The Vote page of the music application will enable the user to vote for his favorite song.

**The Latest Releases Page:**

The Latest Releases page of the music application will display a list of the latest songs released during the last month.

**The Register Page:**

The register page will be used by visitors to register with the application. Visitors will need to provide information such as username, password, address, credit card details during registration.

**The Login Page:**

To logon to the music application, visitors will provide logon information in the Login page. Whether the visitor is a user or administrator is determined based on the user name provided by the visitor. Respective home pages for the users and administrators are displayed.

**User Interface Module:-**

The user interface module contains the following pages:

* The User Home Page
* The Shopping Cart Page
* The Wish list Page
* The Feedback Page
* The Buy Page

**The User Home Page:**

The User Home page is displayed to a user when the user logon to the music application.

**The Shopping Cart Page:**

The Shopping Cart contains the items that a user has selected for buying. The Shopping cart page of the music application will display a list of CDs, Cassettes and other music items selected by the user. Users can add the items to their wish list by clicking the check box next to the items and then clicking on the Add to Wish list button. They can also specify the quantity of the items.

**The Buy Page:**

To purchase music item, a user needs to select songs based on a search criterion. A user can search for items based on song categories and select the songs to purchase on the Buy Page.

**The Wish list Page:**

A user can move the items in the shopping cart to a wish list for later purchase. For example, if the credit limit of a user is exhausted and the user has an item in the shopping cart that he has inclined to purchase, he can put the item on the wish list and purchase the item later. The music application allows the user to view his wish list.

**The Feedback Page:**

To enable users to send feedback to the eMusic World site, there is a feedback page.

**Administrator Module:-**

The administrator Module has the following pages:

* The Delete User Page
* The Add Item Page
* The Modify Item Page
* The Report Page
* The View Wish list Page
* The View Messages Page

**The Delete User Page:**

An administrator can delete a user by using the Delete User page.

**The Add Item Page:**

The add Item page enables administrator to add items to inventory.

**The Modify Item Page:**

This page enables the administrator to manage and modify item details in the inventory.

**The Report Pages:**

The Music application will allow the administrator to view sales reports. The sales reports can be created for the previous week, previous month or for any particular date selected by the administrator.

**The Users Wish list Page:**

This page enables the application developers to view the wish list of the users.

**The View Messages Page:**

The users can send feedback about the web site by using the Feedback Page The administrator need to be able to view these messages so that they can act on the user’s suggestions. Therefore to enable administrators to view feedback from users, a View Message page has been created.

**DATABASE TABLE**

**Table Structure:**

There will be 7 tables in the **MUSIC STORE MANAGEMENT** The normalized form of the table with their structure is described as under: -

**The User Info Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Request ID** | **Int** | **4** |
| **User name** | **Varchar** | **20** |
| **First name** | **Varchar** | **20** |
| **Middle name** | **Varchar** | **20** |
| **Last name** | **Varchar** | **20** |
| **Date of Birth** | **Date time** | **10** |
| **E-mail** | **Varchar** | **25** |
| **Address\_line1** | **Varchar** | **50** |
| **Address\_line2** | **Varchar** | **50** |
| **City** | **Varchar** | **25** |
| **State** | **Varchar** | **25** |
| **Credit card** | **Char** | **16** |
| **Credit card Type** | **Char** | **25** |

**The Feedback Master Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **User name** | **Var char** | **20** |
| **Message** | **Var char** | **50** |
| **Msg date** | **Date time** | **8** |

**The Item Master Table**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data Type** | **Size** |
| **Item Code** | **Char** | **8** |
| **Title** | **Varchar** | **50** |
| **Rate** | **Float** | **50** |
| **Item Desc** | **Varchar** | **50** |
| **Singer** | **Varchar** | **50** |
| **Qty on hand** | **Int** | **50** |
| **Type** | **Varchar** | **50** |
| **Release Date** | **Date time** | **8** |

# 

# The Login Info Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **User name** | **Varchar** | **20** |
| **Password** | **Varchar** | **20** |
| **Secret Question** | **Varchar** | **20** |
| **Secret Answer** | **Varchar** | **20** |
| **Role** | **Char** | **10** |

**The Reply Info Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Reply date** | **Date time** | **8** |
| **Reply msg** | **Varchar** | **50** |
| **Reply user name** | **Varchar** | **20** |

**The Sales Master Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Sale Date** | **Date time** | **8** |
| **Item Code** | **Char** | **8** |
| **Sale qty** | **Int** | **4** |
| **Rate** | **Money** | **8** |
| **User name** | **Varchar** | **20** |

**The Vote Info Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Item Code** | **Char** | **8** |
| **Number votes** | **Int** | **4** |

***PROCESS LOGIC OF EACH MODULE***

**Visitor Module:**

This module has the following pages:

* The Home Page
* The Chart Toppers Page
* The Search Page
* The Vote Page
* The New Release Page
* The Register Page
* The Login Page

**Home Page:**

The Home page of the music application will be the first page to be displayed when a person visits the music application. The page will display a welcome message and, in addition, it will display various menus to facilitate navigation through the application.

**The Chart Toppers Page:**

The Chart Toppers page will display a list of the most popular songs. The popularity of the song will be decided based on the voting at the Vote Page

**The Search Page:**

The Search Page will enable users to perform a search of songs based on the song categories like song name, singer name or album name.

**The Vote Page:**

The Vote page of the music application will enable the user to vote for his favorite song.

**The Latest Releases Page:**

The Latest Releases page of the music application will display a list of the latest songs released during the last month.

**The Register Page:**

The register page will be used by visitors to register with the application. Visitors will need to provide information such as username, password, address, credit card details during registration.

**The Login Page:**

To logon to the music application, visitors will provide logon information in the Login page. Whether the visitor is a user or administrator is determined based on the user name provided by the visitor. Respective home pages for the users and administrators are displayed.

**User Interface Module:-**

The user interface module contains the following pages:

* The User Home Page
* The Shopping Cart Page
* The Wish list Page
* The Feedback Page
* The Buy Page

**The User Home Page:**

The User Home page is displayed to a user when the user logon to the music application.

**The Shopping Cart Page:**

The Shopping Cart contains the items that a user has selected for buying. The Shopping cart page of the music application will display a list of CDs, Cassettes and other music items selected by the user. Users can add the items to their wish list by clicking the check box next to the items and then clicking on the Add to Wish list button. They can also specify the quantity of the items.

**The Buy Page:**

To purchase music item, a user needs to select songs based on a search criterion. A user can search for items based on song categories and select the songs to purchase on the Buy Page.

**The Wish list Page:**

A user can move the items in the shopping cart to a wish list for later purchase. For example, if the credit limit of a user is exhausted and the user has an item in the shopping cart that he has inclined to purchase, he can put the item on the wish list and purchase the item later. The music application allows the user to view his wish list.

**The Feedback Page:**

To enable users to send feedback to the eMusic World site, there is a feedback page.

**Administrator Module:-**

The administrator Module has the following pages:

* The Delete User Page
* The Add Item Page
* The Modify Item Page
* The Report Page
* The View Wish list Page
* The View Messages Page

**The Delete User Page:**

An administrator can delete a user by using the Delete User page.

**The Add Item Page:**

The add Item page enables administrator to add items to inventory.

**The Modify Item Page:**

This page enables the administrator to manage and modify item details in the inventory.

**The Report Pages:**

The Music application will allow the administrator to view sales reports. The sales reports can be created for the previous week, previous month or for any particular date selected by the administrator.

**The Users Wish list Page:**

This page enables the application developers to view the wish list of the users.

**The View Messages Page:**

The users can send feedback about the web site by using the Feedback Page The administrator need to be able to view these messages so that they can act on the user’s suggestions. Therefore to enable administrators to view feedback from users, a View Message page has been created.

**SOFTWARE REQUIREMENT SPECIFICATION**

The **software requirement specification** **(SRS)** is very important part of the software building process, which describes the actual user level requirement from technical point of view. I.e. what the user exactly wants? The objective of preparing the software requirement specification is to represent the requirements of the software in such a manner that ultimately leads to successful software implementation. It is the result of the analysis process of the software development. It should contain all the data the software is going to process, the function it will provide, and the behavior it will exhibit.

This Software Requirements Specifications (SRS) is defined in IEEE Std. 830-1993, IEEE Recommended Practice for Software Requirements Specifications. The synopsis is prepared in the way to fulfill almost all the points needed in S.R.S.

**TESTING PROCESS TO BE USED**

Quality assurance is an important step in software engineering. This overlaps with all the phases of development right from the requirement analysis. This quality requirement of the software system must be clearly extracted during the requirement analysis and all the subsequent phases should be made biased to that, the final testing will become trivial and less expensive.

There are number of quality parameters like correctness, accuracy, reliability, robustness, efficiency, effectiveness, reusability, maintainability etc.. The state of requirement of each of these parameters will vary depending upon the name and domain of the application. The testing should be done at the end of all development steps. Even though the final testing and verification are inevitable for better life and functionality of the software.

The different software testing approaches and methods like white box testing and black box testing. The major phases in testing are design of test plan, setting up test case and test candidate and test procedure, testing and correction. This is a cycle process and the software will circulate through all the steps till it attends the required quality. The testing is carried in the following steps.

**Unit testing**

Unit testing focuses verification effort on the smallest unit of software design the module. Using the details design description as a guide, important control paths are tested to uncover errors within boundary of the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing.

Unit testing is normally considered an adjacent to coding steps. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins. A review of design information provides guidance for establishing test cases that are likely to uncover error in each case of the categories discussed above. Each test case should be coupled with a set of expected results.

**Integration testing**

Integration testing is systematic technique for constructing the program structure while at the same time conducting test to uncover error associated with interfacing .The objective is to take unit tested modules and build a program structure that has been dictated by design.

There is often a tendency to attempt no incremental integration; that is to construct the program using a “big bang “approach. The entire modules are combined in advance. The entire program is tested as whole and chaoses usually result! A set of error is encountered. Correction is difficult because the isolation of cause is complicated by the vast expanse of entire program. Once errors are corrected, new ones appear and process continues in a seemingly endless loop.

**SECURITY MECHANISMS**

**TESTING & VALIDATION**

The approach of web app testing adopts the basic principle for all software testing and applies a strategy and tactics that have been recommended for object oriented system .the following steps summarizes the approach:

1. T**he content model for the web app is reviewed to uncover errors.**

This is like copy-editing.

**2. The design model for the web application as reviewed to uncover**

**navigation error.**

Use cases derived as part of the analyst activity, allow a web Engineer exercise each usage scenario against the architectural and navigational design. These non-executable test help uncover error in navigation.

3. **Selected processing component and web page is unit tested**.

When web apps are considered, the concepts of the unit changes, each web page encapsulated in itself content navigation link as well as script, form and applet (processing element). It is not always possible or practical to test each of these characteristic individually.

4. **The architecture is constructed and integration tests are conducted**.

The strategy for integration testing depends upon the architecture that has been chosen for the web application.

5**. The assembled web application is tested for overall functionality and content delivery**.

Like conventional validation, the validation of web based systems and application focuses on user visible action and user recognizable output from the system. To assist in the derivation of validation tests the tester should draw upon use cases the use cases provides a scenario that has high likelihood of uncovering errors in user interaction requirement

.

6. **The web application is implemented in a variety of different environmental configuration and is tested for compatibility with each configuration.**

7. **The web application is tested by controlled and monitored population of end user.**

Finally with the modular concept inside the application it is being also tested for its Reliability:

The system reliability will be insured through data integrity rules built into the database at the backend and the system rules built into the front-end application. The system will take assurance from the user before making any changes permanent.

**MAINTAINABILITY:**

The system has been designed taking care of modularity. Faults in the system can be traced to modules.

**VALIDATION CHECKS**

This will be as such to maintain consistent and persistent information on the web when most of the time the project has to deal with uploads so a minor error will down the impression of the company.

**Scope for Future Enhancement**

As changes are always necessary in future it applies to software development also but these changes should be appreciable in nature. These appreciable changes will make the software to fight for its survival in the competitive market. Hence it is necessary to think about the future enhancements at present.

The system ‘MUSIC STORE MANAGEMENT’ will fulfill the entire requirement of the clients. The system is developed according to the present requirements of the company. The system is developed as easy as possible for the sake of end users.

One drawback of my system is that the client cannot view, search and purchase music according to a particular language option .By the next time I would like to add this facility.

By the next time I would like to add two more modules: Purchase Module and Accounting Module. Purchase Module deals with purchasing activities of music related items. Accounting Module deals with all accounting activities such as billing, ledger preparation, balance sheet preparation, profit and loss account preparation etc.

In the present system transaction is through a particular bank or through money orders. In future I would like to make it through credit cards. Credit card validation techniques are needed for that.

The developed software for the organization is flexible and it can be made to run on all kinds of platforms. The system is error free and highly portable. It can be implemented in any servers in the Internet providing an easy access to the clients. It also has more options of the future developments.

**LIMITATION OF THE PROJECT**

* Can’t cover the financial data of the company.
* Can’t cover the windows interface with this project.

**Bibliography**

**Serial No. Title**

**1. Servlet Programming 2. Java Server Programming**

**3. Java Server Pages**

**4. Oracle8i**

**A. Online Resources**

* + - **http://www.google.com**
    - **http://oracle.com**
    - **http://Redbooks.IBM.com**
    - **http://javaguru.com**
    - **http://www.sun.com**

***R***

***EPORT***

***MUSIC STORE MANAGEMENT***



# PROJECT DESCRIPTION

This is portal based automation project, which provides communication between the various users for music items such as music CD’s/Cassettes. User can purchase their needed music item anywhere, anytime just clicking their mouse. This project increases the interest of purchasing. User can easily finished their shopping through this shopping cart.

Screen design/Graphical User Interface:-

Graphical User Interface (GUI) that is straightforward and easy to navigate has been designed. This GUI provide various screens with appropriate incorporate icons, hyperlinks etc. to facilitate screen navigation and data entry. The user can easily add items to their cart, and they can easily remove items form the cart if they needed. The user has the ability to return to home page from any location within the application.

The following GUI form for user interaction can be conferred in the **“MUSIC STORE MANAGEMENT”** system

The **“MUSIC STORE MANAGEMENT”** is divided into three modules:

* + **Visitor module**
  + **User module**
  + **Administration module**

**REQUIREMENT SPECIFICATION**

**HARDWARE SPECIFICATION**

Processor : Pentium-III

Memory : 128MB

Hard Disk : 20GB

Floppy Drive : 1.44MB

Monitor : 14’’ or above

Mouse : Standard Mouse

Key Board : 104 keys

**SOFTWARE SPECIFICATION**

Operating System  **:**  WINDOWS 2000 Server

Programming Language  **:** JAVA

Server Technologies  **:**  J2EE (JSP, SERVLETS, JDBC)

Application Server  **:** Jakarta Tomcat 4.1

Database Support  **:** ORACLE 8i

**DEVELOPMENT ENVIORNMENT**

###### An Over View of J2EE:-

By using **J2EE** technology the Internet banking facilitates to know account details, apply loan online, money transfer, balance enquiry-Pay and value added services.

* **J2EE** applications are made up of components. A *J2EE component* is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and that communicates with other components. The J2EE specification defines the following J2EE components:
* Application clients and applets are components that run on the client.

Introduction to HTML:-

This specification defines the HyperText Markup Language (HTML), the publishing language of the World Wide Web. This specification defines HTML 4.01, which is a subversion of HTML 4.

In addition to the text, multimedia, and hyperlink features of the previous versions of HTML. HTML 4 supports more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to users with disabilities. HTML 4 also takes great strides towards the internationalization of documents, with the goal of making the Web truly World Wide.

**HTML essintials:**

1. Web Browser
2. Tags
3. Web Browser Support
4. HTML version

**Web Browser:**

Microsoft Internet Explorer and Netscape Navigator is currently thae most popular web browser. Microsoft Internet Explorer is included with windows & windows 2000.

**Tags:**

Tags tell web browsers about the structure of web page. Each tag gives a specifc instruction and is surrounded by angle bracket <>. Most tags have an opening tag and closing tag.

**Web Browser Support:**

A web browser may not look the same when displayed in different web browsers. Not all web browsers support all the features of HTML and each browser may interpret HTML tags differently . some companies that make web browsers have also developeed their own tags that web browsers made by other companies may not be able to understand. If a web broser does not understand a tag the tag is usually ignored.

**HTML Version:**

There are several versions of HTML. HTML specification are constantly evolving and a new version of HTML is realeased in every 2 years. HTML version 4.01 is latest version of HTML.

**Introduction to Java Script:-**

JavaScript is a compact, object-based scripting language for developing client and server Internet applications. Netscape Navigator interprets JavaScript statements embedded in an HTML page, and Livewire enables you to create server-based applications similar to Common Gateway Interface (CGI) programs. This book describes the JavaScript language and its use in Navigator. For information on developing server-based JavaScript applications, see the *Livewire Developer's Guide.*

Client-side JavaScript statements embedded in an HTML page can respond to user events such as mouse-clicks, form input, and page navigation. For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, the HTML page with embedded JavaScript can check the entered data and alert the user with a dialog box if the input is invalid.

<SCRIPTLANGUAGE="*JavaScriptVersion*">  
 JavaScript*statements.*..  
</SCRIPT>

Since browsers typically ignore unknown tags, non-JavaScript-capable browsers will ignore the beginning and ending SCRIPT tags. All the script statements in between are enclosed in an HTML comment, so they are ignored too. Navigator properly interprets the SCRIPT tags and ignores the line in the script beginning with the double-slash (//).

**Introduction to JSP:-**

Java Server Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server- and platform-independent.

In Java 2 Platform, Enterprise Edition (J2EE) v1.4, JSP technology has simplified the page and extension development models with the introduction of a simple expression language, tag files, and a simpler tag extension API, among other features. This makes it easier than ever for you to build pages based on JSP technology.

JSP technology is focused on simplifying the generation of dynamic web content. The JSP 2.0 specification (JSR-152) substantially extended the technology by integrating a simple yet powerful expression language, simplifying the tag extension API, and enhancing the pure XML syntax, among other important enhancements. These enhancements greatly reduced the learning curve of the technology, warranting a major version number upgrade.

The scope of JSP 2.1 is much narrower and focuses on better alignment with the next release of Java Server Faces. Java Server Faces 1.0 (JSR-127) defines a standard framework for building User Interface components, and builds on top of JSP 1.2 technology. Because JSP 1.2 technology does not have an integrated expression language and because the JSP 2.0 EL does not meet all of the needs of JSF, a new expression language was developed for JSF 1.0. The JSF EG attempted to make the language as compatible with JSP 2.0 as possible but some differences were necessary. The JSF EG agreed that the JSF 1.0 EL will be phased out in the next release of the JSF specification in favor of the JSP 2.1 expression language. It is a goal, therefore, of JSP 2.1 to enhance the expression language to meet the needs of JSF technology. Many of these enhancements are likely to be useful in other contexts as well.

**Advantages of JSP over other Server Side Technologies:**

* **Vs. Active Server Pages (ASP).** ASP is a similar technology from Microsoft. The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS-specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.
* **Vs. Pure Servlets.** JSP doesn't give you anything that you couldn't in principle do with a servlet. But it is more convenient to write (and to modify!) regular HTML than to have a zillion println statements that generate the HTML. Plus, by separating the look from the content you can put different people on different tasks: your Web page design experts can build the HTML, leaving places for your servlet programmers to insert the dynamic content.
* **Vs. Server-Side Includes (SSI).** SSI is a widely-supported technology for including externally-defined pieces into a static Web page. JSP is better because it lets you use servlets instead of a separate program to generate that dynamic part. Besides, 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. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server-side resources like databases, catalogs, pricing information, and the like.
* **Vs. Static HTML.** Regular HTML, of course, cannot contain dynamic information. JSP is so easy and convenient that it is quite feasible to augment HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

**Introduction to Servlet:-**

Java Servlet technology provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. A servlet can almost be thought of as an applet that runs on the server side--without a face. Java servlets make many Web applications possible.

Java servlets are more efficient, easier to use, more powerful, more portable, and cheaper than traditional CGI and than many alternative CGI-like technologies. (More importantly, servlet developers get paid more than Perl programmers :-).

* **Efficient:** With traditional CGI, a new process is started for each HTTP request. If the CGI program does a relatively fast operation, the overhead of starting the process can dominate the execution time. With servlets, the Java Virtual Machine stays up, and each request is handled by a lightweight Java thread, not a heavyweight operating system process. Similarly, in traditional CGI, if there are *N* simultaneous request to the same CGI program, then the code for the CGI program is loaded into memory N times. With servlets, however, there are *N* threads but only a single copy of the servlet class. Servlets also have more alternatives than do regular CGI programs for optimizations such as caching previous computations, keeping database connections open, and the like.
* **Convenient:** Hey, you already know Java. Why learn Perl too? Besides the convenience of being able to use a familiar language, servlets have an extensive infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, tracking sessions, and many other such utilities.
* **Powerful:** Java servlets let you easily do several things that are difficult or impossible with regular CGI. For one thing, servlets can talk directly to the Web server (regular CGI programs can't). This simplifies operations that need to look up images and other data stored in standard places. Servlets can also share data among each other, making useful things like database connection pools easy to implement. They can also maintain information from request to request, simplifying things like session tracking and caching of previous computations.
* **Portable:** Servlets are written in Java and follow a well-standardized API. Consequently, servlets written for, say I-Planet Enterprise Server can run virtually unchanged on Apache, Microsoft IIS, or Web Star. Servlets are supported directly or via a plug in on almost every major Web server.
* **Inexpensive:** There are a number of free or very inexpensive Web servers available that are good for "personal" use or low-volume Web sites. However, with the major exception of Apache, which is free, most commercial-quality Web servers are relatively expensive. Nevertheless, once you have a Web server, no matter the cost of that server, adding servlet support to it (if it doesn't come preconfigured to support servlets) is generally free or cheap.

**Introduction to JDBC:-**

JDBC technology is an API (included in both J2SE and J2EE releases) that provides cross-DBMS connectivity to a wide range of SQL databases and access to other tabular data sources, such as spreadsheets or flat files. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment

JDBCTM was designed to keep simple things simple. This means that the JDBC API makes everyday database tasks, like simple SELECT statements, very easy. This trail will walk you through examples of using JDBC to execute common SQL statements, letting you see for yourself how easy it is to use the basic JDBC API.

The first thing you need to do is establish a connection with the DBMS you want to use. This involves two steps: (1) loading the driver and (2) making the connection.

### Loading Drivers:

Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

Class. For Name ("sun.jdbc.odbc.JdbcOdbcDriver");

The second step in establishing a connection is to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea:

Connection con = DriverManager.getConnection (url, "my Login", "my Password")

**Introduction to Oracle:-**

With Oracle Database, the first relational database designed for Grid Computing, your information is securely consolidated and always available. Oracle Database 10g has the lowest total cost of ownership by making the most efficient use of hardware and IT resources. Oracle is the best choice for large enterprises, small and midsize businesses, and departments alike.

Globalization, Simplification, Standardization, Automation, Innovation. These five principles underlies very thing we dot Oracle. Oracle Services help you get the most from your technology investment. Working with you every step of the way, Oracle provides a variety of services that span the complete solution life cycle. Whether you need consulting, financing, outsourcing, support or education, you can get it from the experts who know Oracle best.

**Connecting the Database we have follow the following steps:**

1. Loading the Driver

Class. For Name (“sun.jdbc.odbc.JdbcOdbcDriver”);

1. Create the Connection object.

Connection con = Driver Manager.GetConnection(“jdbc:odbc:dsn”);

1. Create Statement object

Statement stmt = con.createStatement();

1. Execute the SQL query

ResultSet rs=Stmt.executeQuery(“select \* from login”);

1. Navigate through the records

While rs.next()

1. Close the Connections

Con.close();

**Introduction to Web server:-**

Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process.

Tomcat is developed in an open and participatory environment and released under the Apache Software License. Tomcat is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved.

**Tomcat 4.1** Tomcat 4.1 is the current focus of development. While it supports the same Servlet and JSP Specification versions as Tomcat 4.1, there are significant changes in many areas under the hood, resulting in improved performance, stability, and total cost of ownership. Please refer to the Tomcat 4.1 Change log for details.

**SYSTEM ANALYSIS**

**Existing system:-**

Music and music items has become an inevitable part of our life. Music is one of the greatest so others and healers of an afflicted heart. Customers obtain these music items from music stores. An existing system is nothing but a manual Music store. A manually run Music store has so many drawbacks. In such systems the customer will go to such music shops and search for a specific music CD/Cassette. If the music CD/Cassette is available then only they will purchase the same. This system has its own drawbacks.

* Customer has to manually visit the music shop and need to purchase the music item they needed.
* The wastage of time for searching a particular venue and a particular shop.
* The wastage of money as transportation for searching a particular shop.
* Manual activities involve ground work, this involves more monetary terms.

**Proposed system:-**

The proposed system uses GUI framework. This system is highly user friendly because the entire programs are menu driven so that the new comer can use the software efficiently. It could be necessary to make corrections in the program depending on the changes in the system specification in future.

This product has been mainly designed to overcome some of the problems faced with the manual system. The main problem faced was unnecessary delay. The previous system in use was also expensive and time consuming. In order to avoid unnecessary delay and minimize the flaws that existed in the previous system a follow up module for the existing system has been designed called the **‘MUSIC STORE MANAGEMENT’**.

The main intention of the proposed and designed system is to automate the shopping channel between the company and the clients, and easy the user work. Through this an end user can easily purchase the needed thing from his home, or anywhere by just clicking the mouse. And it increases the mentality of shopping. This system is designed to avoid ineffective and inefficient customer service that has been recognized as the major area of focus.

**Feature of proposed system:-**

* The application will enable visitors to register with sites.
* The application will enable visitors to perform activities such as search for music items, vote for popular music and view latest releases and chart busters.
* The application will enable a registered user to select items such as CD’s and Cassettes for purchase using the shopping cart.
* The application will have an administration page, which will enable administrator to maintain user detail and manage music items in inventory.
* The application will allow users to search for music based on multiple keywords such as album name, music category or singer name.

**Advantages of proposed system:-**

* Automate the communication between the users and the company.
* Save the time of customer.
* It provides 24 hours of facility.
* Everyone can access the system those who are living in any place.
* Easy the work of shopping.
* Increase the shopping interest.
* Decrease the time wastage.

**IDENTIFICATION OF THE NEED**

The Identification of Need is one of the most difficult tasks of System Analysis i.e. developing a clear, in-depth under – standing of the problem being investigated, without which it becomes impossible to specify the requirements for a new Project with an accuracy.

While Identifying the Needs for this Project I have kept the following things in my mind:

* The Performance of the System.
* The information being supplied and its form.
* The economy of processing.
* The control of the information Processing.
* The efficiency of the existing System.
* The security of the Data and Software.
* The security of the equipment and personnel, etc.

After identification of Need, it is defined and a general direction or method for solving is also determined. Then Project boundaries are defined.

**PRELIMINARY INVESTIGATION**

The First Step in the System Development Life Cycle is the Preliminary Investigation to determine the feasibility of the system.

The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of the information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

**In my Project the following Objectives of the Preliminary investigation is accomplished:**

* Clarify and understand the project request.
* Determine the size of the project.
* Assess costs and benefits of alternative approaches.
* Determine the technical and operational feasibility of alternative approaches.
* Report the findings to management; with recommendations outlining the acceptance or rejection of the proposal.

# CONDUCTING THE INVESTIGATION

The data that the analysts collect during preliminary investigations are gathered through three primary methods: reviewing organization documents, on-site observations and conducting interviews.

**REVIEWING ORGANIZATION DOCUMENTS**

The analyst conducting the investigation first learn about the organization involved in, or affected by the project. In Bank Automation System, it is necessary to know how the Bank works and who are the persons directly associated with process / working of the Bank.

**ON-SITE OBSERVATION**

Another important technique to collect data is on-site observation. The purpose of the On-site observation is to get as close as possible to the real system being studied. During On-site Observation, office environment can be seen, work load, method of work and facilities provided by the organization to the users can be studied.

**CONDUCTING INTERVIEWS**

Interviews allows to learn more about the nature of the Project request and reasons for submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally and technically.

**FEASIBILITY STUDY**

All projects are feasible when given unlimited resources and infinite time. Its both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. The efforts and resources spent in developing the system will be a waste if the end solution does not offer timely and satisfactory solution to its users.

Feasibility study is a test of system proposed regarding workability, impact on the organization ability to meet user needs, and effective use of resources. Thus when a new application is proposed, it normally goes through a feasibility study before it is approved for development.

Feasibility and risk analysis are related in many ways. If project risk is great, the possibility of producing quality software is reduced.

**In this Project I have conducted the three aspects of Feasibility Study.**

**TECHNICAL FEASIBILITY**

The Technical Feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

* The facility to produce outputs in a given time.
* Response time under certain conditions.
* Ability to process a certain volume of transaction at a particular speed.
* Facility to communicate data to distant location.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should given the complete picture about the system’s requirements like how many workstations are required, how these units are interconnected so that they could operate and communicate smoothly. What speeds of input and output should be achieved at particular quality of printing. This can be used be used as a basis for the tender document against which dealers and manufactures can later make their equipment bids. Specific hardware and software products can then be evaluated keeping in view with the logical needs.

At the feasibility stage it is desirable that two or three different configurations will be pursued that satisfy the key technical requirements but which represent different levels of ambitions and cost. Investigation of these technical alternatives can be aided by approaching a range of suppliers for preliminary discussions. Out of all types of feasibility, technical feasibility generally is the most difficult to determine.

**OPERATIONAL FEASIBILITY**

It is mainly related to human organizational and political aspects. The points to be considered in this Project related to Operational Feasibility are as follows:

* What changes will be brought with the system?
* What organizational structures are distributed?
* What new skills will be required? Do existing staff members have these skills? If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations. This feasibility study is carries out by a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and are skilled in system analysis and design process.

As far as this project is concerned the changes which we have to be brought depends upon whether we are going to build a new project or we going to modify some establishment in the late one first of all we have make a network environment i.e establishment of a server is must. Then our focus goes towards workstations. Keeping in view of their hardware requirements like network interface card etc.

**Regarding this project**, distribution of organizational structures is also essential because of security concerns, as there are different departments having their particular tasks I have already mentioned earlier like a system administrator should have the authentication to provide different access permission to its clients.

**ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis; the procedure is to determine the benefits and savings that expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

This feasibility also depends upon quality of staff hired and the proposed duration of time taken in this project sometimes it might be possible due to extension of time duration may fall the project under loss. The study of feasibility changes from phase to phase of the project development.

In this project although this feasibility study doesn’t matter much in the case new setup of project because we start according to client specification but on the other hand if we have to modify over existing system we must take care of our existing resources and must analyse specially the working condition of hardware like server quality etc.

**SOFTWARE ENGINEERING PARADIGM APPLIED**

To solve actual problems in an industry setting, a software engineer or a team of engineers must incorporate a development strategy that encompasses the process, methods, and tools layers. The strategy is often referred to as a process model or a software-engineering paradigm. A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required.

Numbers of paradigm available

* The Linear Sequential Model
* The Prototype Model
* The RAD Model
* The incremental Model
* The Spiral Model
* The WINWIN Spiral Model
* The Component- based development Model
* The Concurrent Development Model
* The Formal Methods Model
* The Fourth Generation Techniques Model

In this project ‘ONLINE DEALS IN’ I have implemented the Linear Sequential Model.

THE LINEAR SEQUENTIAL MODEL

It is also called as the classic life cycle or the Waterfall Model, the linear sequential model suggests a systematic, sequential approach to software development that begins at the system level and progresses through Analysis, Design, Coding, Testing and Support.

Modeled after a conventional engineering cycle, the linear sequential model encompasses the following Activities.

# Test

# Code

# Design

# Analysis

* System/ Information Engineering and Modeling.
* Software Requirement Analysis.
* Design.
* Code Generation.
* Testing.

**SOFTWARE REQUIREMENT SPECIFICATIONS**

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional description, a representation of system behavior, an indication of performance requirement and design constraints, appropriate validation criteria, and other information pertinent to requirement.

* The **Introduction** to software requirements specification states the goals and objectives of the software, describing it in the context of the computer based system.
* The **Information Description** provides a detailed description of the problem that the software must solve. Information content, flow, and structure are documented.
* A Description of each function required to solve the problem is presented in the **Functional Description.**
* **Validation Criteria**is probably the most important and, ironically, the most often neglected section of the Software requirement Specification.

**In this Project the Software Specifications are used for different purposes. Here are these major uses.**

**STATEMENT OF USER NEEDS**

A main purpose of the product specification is to define the need of the product’s user. Some times, the specification may be a part of a contract sign between the producer and the user. It could also form part of the user manuals. A user’s needs are sometimes not clearly understood by the developer. If this is the case, a careful analysis – involving much interaction with the user should be devoted to reaching a clear statement of requirements, in order to avoid possible misunderstandings.

Sometimes, at the beginning of a project, even the user has no clear idea of what exactly the desired product is. Think, for instance, of user interfaces. A user with no previous experience with computer products may not appreciate the difference between, say, menu driven interaction and a command line interface. Even an exact formulation of system functions and performance may be missing an initial description produced by an inexperienced user.

**A STATEMENT OF THE REQUIREMENTS FOR THE IMPLEMENTATION**

Specifications are also used as a reference point during product implementation. In fact, the ultimate goal of the implementation is to build a product that needs specification. Thus specifications are used by the implementers during design to make design decisions and during the verification activity to check that the implementation compiles with the specifications.

**DESIGN**

The Designing of a Software is first of three technical activities – design, Code generation, and test – that are required to build and verify the software. Each activity transforms information in a manner that ultimately results in validated computer software.

The Design task produces a data design, an architectural design, an interface design and component design.

The Design of an information system produces the details that clearly describes how a system will meet the requirements identified during system analysis. The system design process is not a step by step adherence of clear procedures and guidelines. When I started working on system design, I face different types of problems, many of these are due to constraints imposed by the user or limitation of hardware and software available. Some time it was quite difficult to enumerate the complexity of the problems and solutions thereof since the variety of likely problems is so great and no solutions are exactly similar however the following consideration I kept in mind during Design phase.

### DESIGN OBJECTIVES MADE IN THIS PROJECT

The primary objective of the design is to deliver the requirements as specified in the feasibility report. These are the some of the objective, which I kept in mind.

**PRACTICALITY**

The system is quite stable and can be operated by the people with average intelligence.

**EFFICIENCY**

I tried to involve accuracy, timeliness and comprehensiveness of the system output.

**COST**

It is desirable to aim for the system with a minimum cost subject to the condition that it must satisfy all the requirement.

**FLEXIBILITY**

I have tried that the system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.

**SECURITY**

This is very important aspect which I followed in this designing phase and tried to covers the areas of hardware reliability, fallback procedures, physical security of data.

**DATA FLOW DIAGRAMS FOR THE PROJECT**

0 LEVEL D.F.D.:-

This is the context level D.F.D. of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The purposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system.

Context Level Diagram

Valid

communication

Services

User

User

**0 LEVEL D.F.D. FOR “MUSIC STORE MANAGEMENT”**

1ST LEVEL D.F.D.:-

This D.F.D. shows all the processes together with all the data stores (tables). It shows the true data flow i.e. how data is actually flowing in the system. Data is coming from which table and going into which table is clearly shown by this DFD. This DFD is the main reference for the development of the system. After understanding the whole system, the application developer will fall back upon this DFD during the Development phase.

Valid

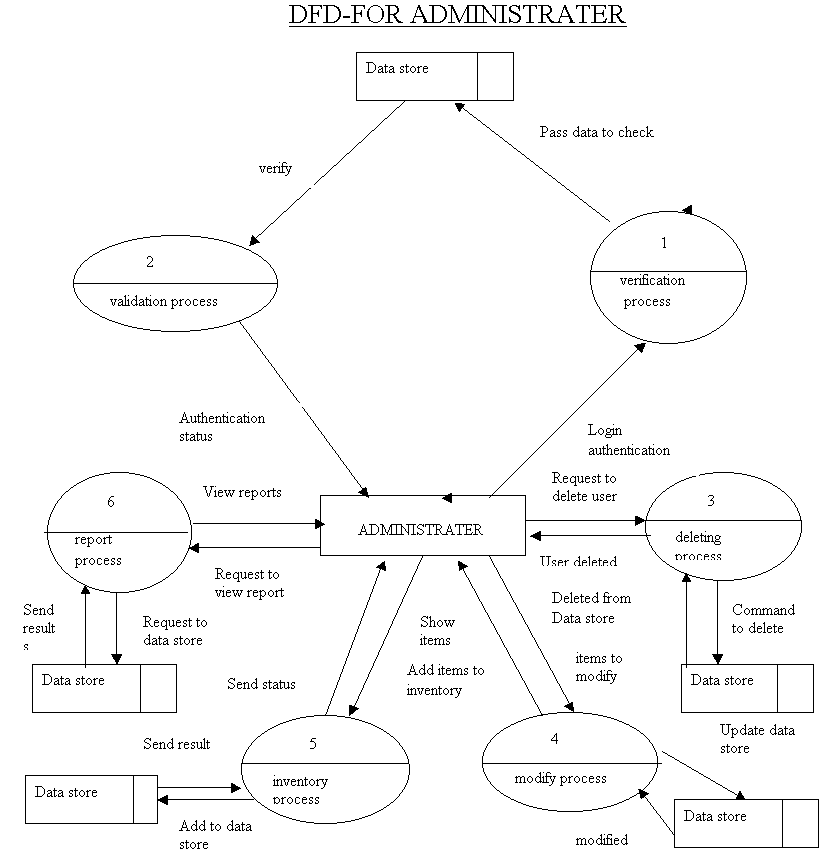
Login

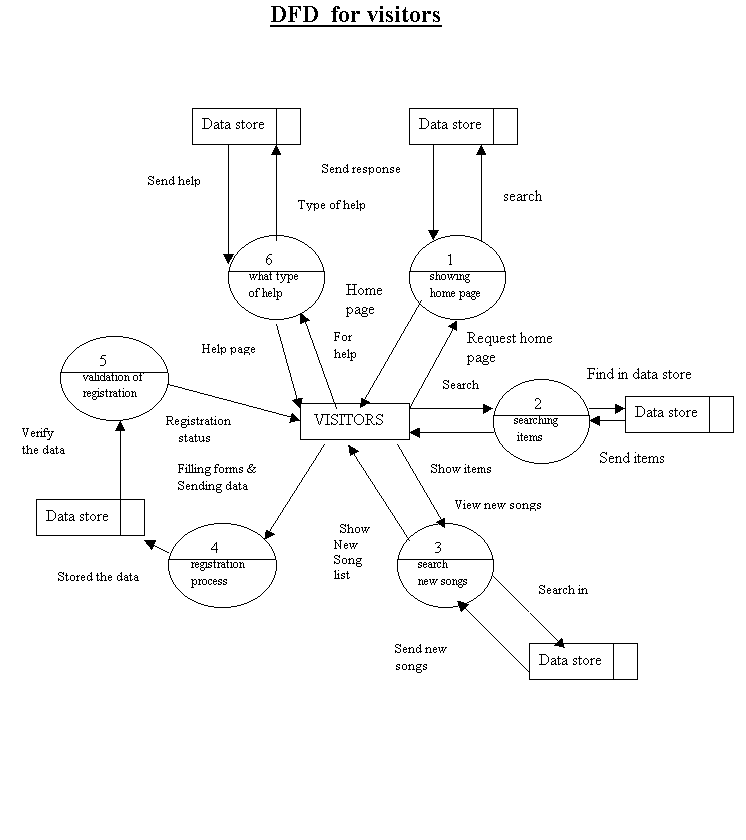
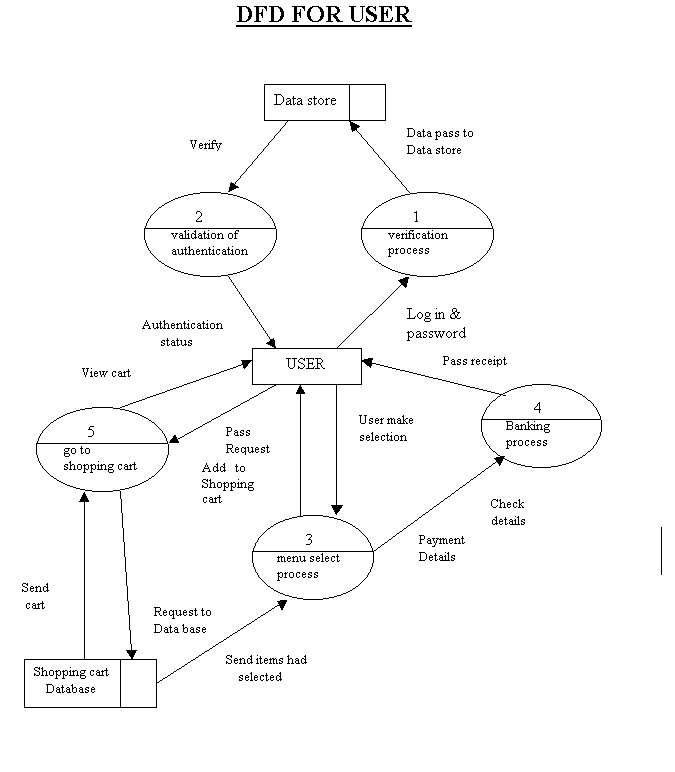
Valid

User

Invalid

Valid

******



**ER-Diagram**

Enter

Login

Information

Feed back

Master

Reply

User information

Registration

Purchases

Reply info

Sales master

Vote

Vote info

Item master

**SYSTEM DESIGN**

Sale info

**Design Technique:**

Design is a multi steps process that focuses on data structure, software, software architecture, external details and interface between the modules. The design processes also translate the requirements into representation of software that can be accessed for quality before coding begins.

Computer software designs changes continually as new methods, better analysis and broader understanding evolve. Software design is at a relatively early stage in its revolution. Therefore, software design methodology locks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines.

How ever techniques for software design do exist, criteria for design qualities are available and design notation can be applied. Once software requirements have been analysed and specified, software design is the first of three activities- Design, code, test, that are required to build and verify software.

Each activities transform information in a manner that ultimately results in a validation of computer software. The importance software design can be started with a single word quality. Design is the place where quality fostered in software development. Design provides us with the representations of the software that can be accessed for quality.

Design the only way that we can accurately translate a customer’s requirement into a finished software product or system. Without design, risk of building an unstable system exists-one that will fail when small changes are made one that may be difficult to test.

**Internal design**

The input design is the link between the information system and the users. It comprises the directing specification and procedures for data preparations and those steps that are necessary to put transaction data into a usable form for processing data entry.

The designs of inputs focuses on controlling the amount of inputs required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple.

System analyst decides the following input designs details:

* Why data to input?
* What medium to use?
* How the data should be arranged or coded?
* The dialogue to guide users in providing input.
* Methods for performing input validation and steps to follow when error occurs.

Several activities have to be carried out as part of the overall input process. They include some or all of the following stages

* Data recording (that is, collection of data at its source);
* Data transcription (that is, transfer of data to an input form);
* Data conversion (that is, checking the conversion);
* Data control (that is, checking the accuracy and controlling the flow of the data to the computer);
* Data transmission (that is, transmitting or transporting the data to the computer);
* Data validation (that is, checking the input data by program when it enters the computer system);
* Data correction (that is, correcting the errors that are found at any of the earlier stages).

**External design**

Designing computer output should proceed in an organized, well thought out manner. The term output applies to any information produced by an information system whether printed or displayed. When analyst designs computer output, they identified the specific output is needed to meet the information requirements.

Computer output is the most important and direct source of information to the user. Output design is a process that involves designing necessary outputs that have to be various users according to their requirements.

Efficient intelligent output design should improve the systems relationship with the users and help in decision-making. Since the reports are directly required by the management for taking decisions and to draw conclusions, they must be designed with utmost care and the details in the records must be simple, descriptive and clear to the user.

The options for the outputs and reports are given in the systems menu. When designing output, system analyst must accomplish the following:

* Determine the information to present.
* Decide whether to display or print the information and select the output medium.
* Arrange the present of information acceptable format.
* Decide how to distribute the output to intended receipts.

**Architectural design**

Architectural design begins with recognition that the screen is composed of different areas. Layout tools assist the analyst in specifying the content of the single and multiple design formats. All screens have been provided with menus, push buttons facilities, icons and control buttons such as add/delete/edit/find/clear /exit etc.

The main screen consists of main menu from which we can move to another forms or screens.

In designing output screens we need area for:

* Heading and titles.
* The content of display.
* Message and instruction.
* Sometimes explanations for information in the reports

**Procedural Design**

The procedural design transforms structural component in to a procedural description of the software. Source is generated and testing is conducted to integrate and validate to software. The design of input and output screen comes under the procedural design input/output design is according to needs of the user.

The input and output design are related to each other in sense that the accuracy data depends on the accuracy of the input data and processing of input data. Thus for this proposed system the input and output design are in the form of forms. In the forms based interface design the user give the input by filling the blanks of the screen.

**Database design**

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

The objectives of the file design are to provide effective auxiliary storage and to contribute to the overall the efficiency of the computer program component of the system. In concepts of database design, there are two types of data – physical data and logical data.

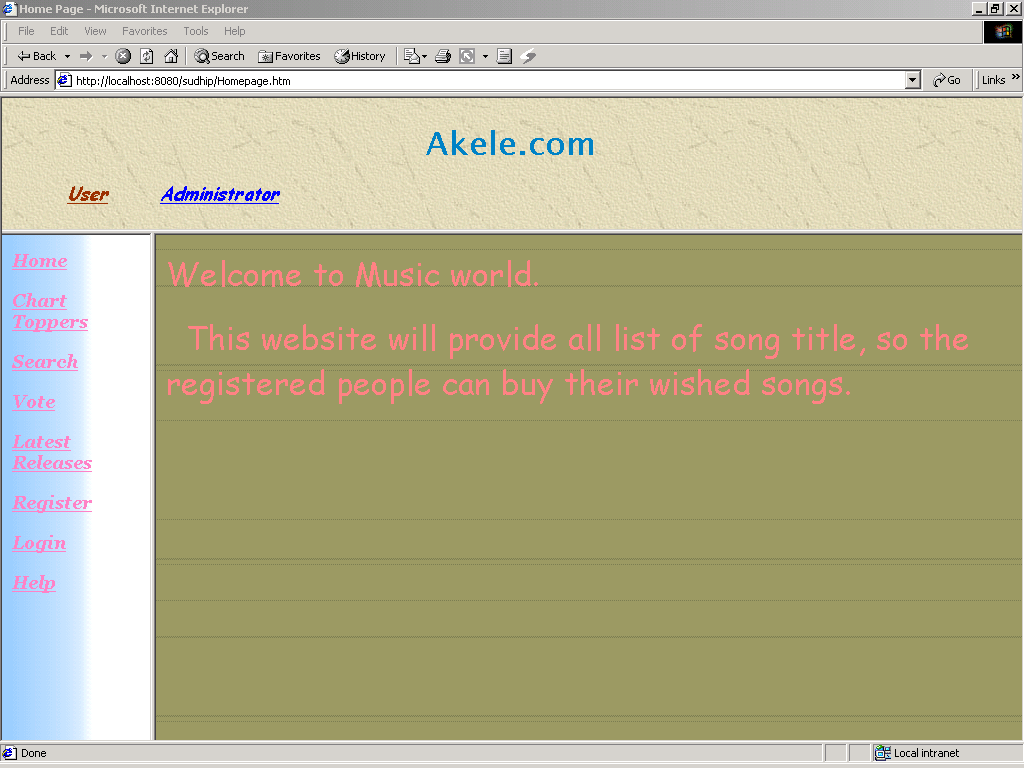
Physical data is that which is written on those pieces of paper. Logical data are those, which are calculated based on some of the retrieved data in a certain sequence in summary form. In a computer-based data processing system, separation of physical and logical data provides the same advantages.

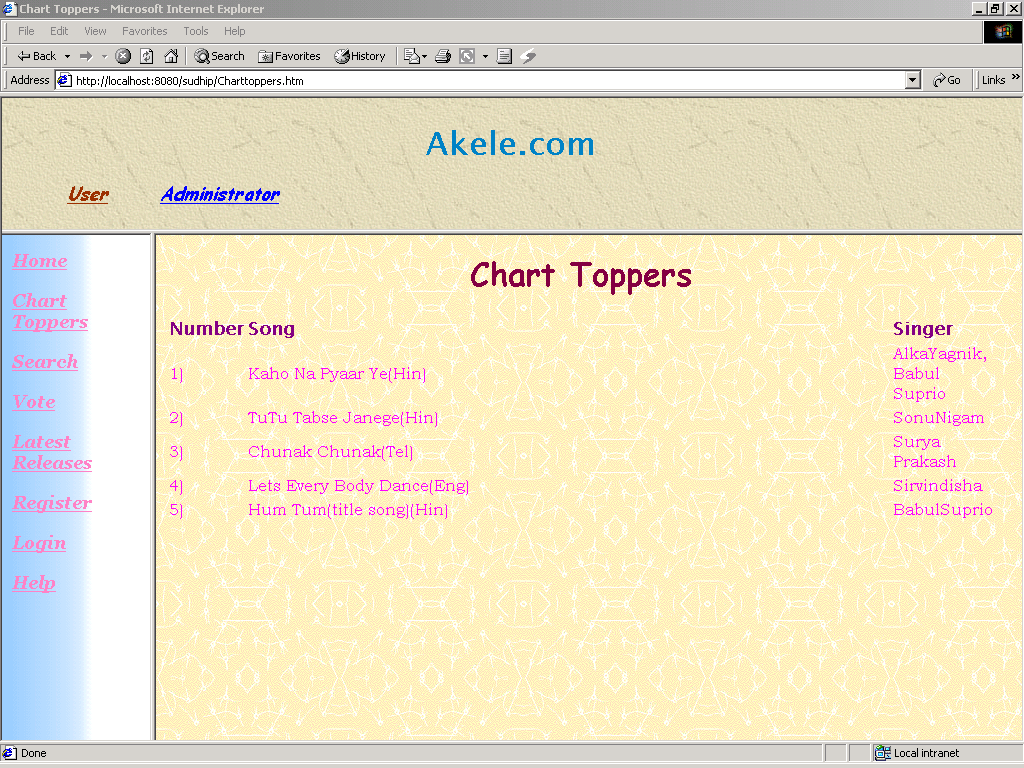
**MUSIC STORE**

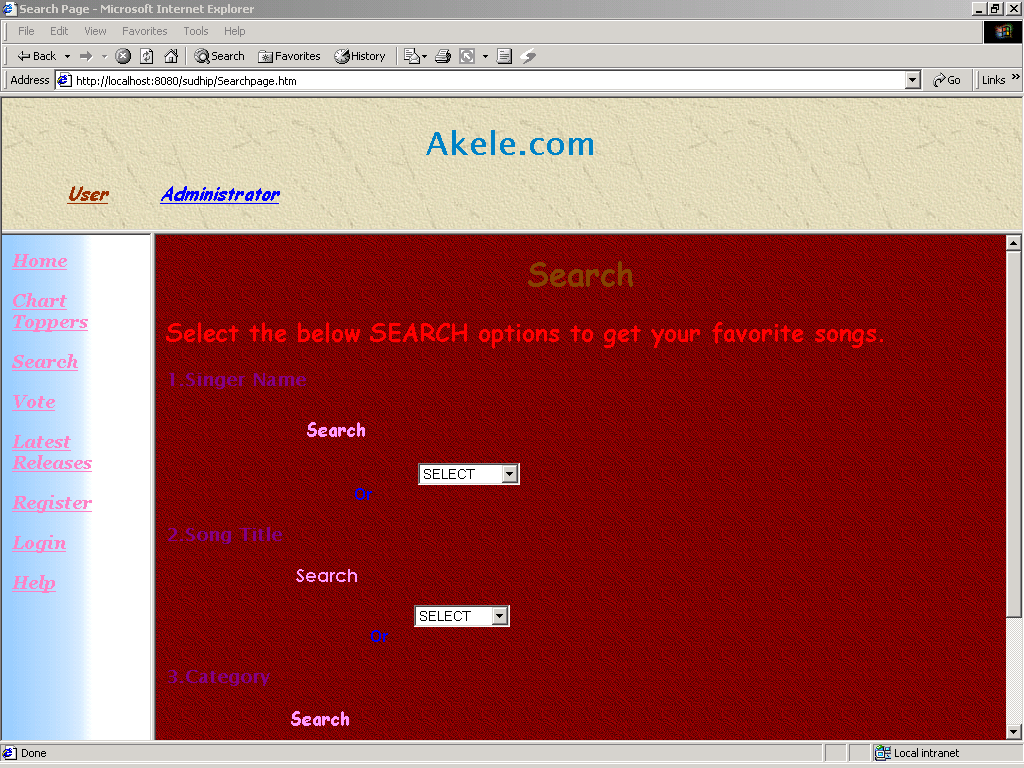
**MANAGEMENT**

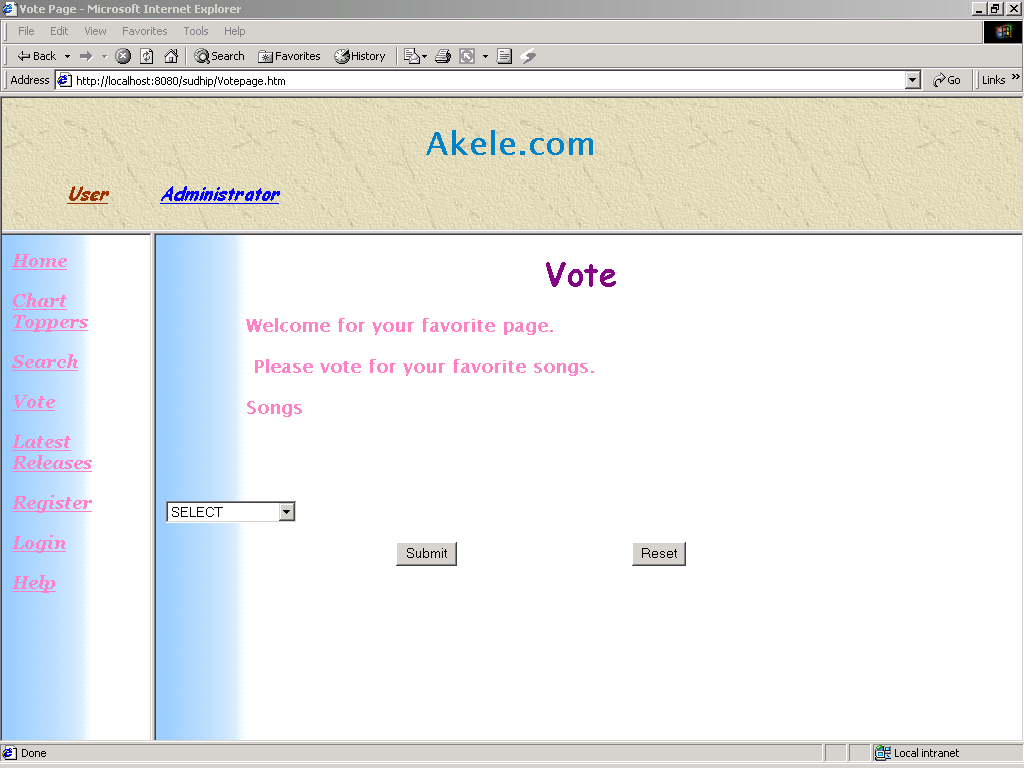
Screen Shots

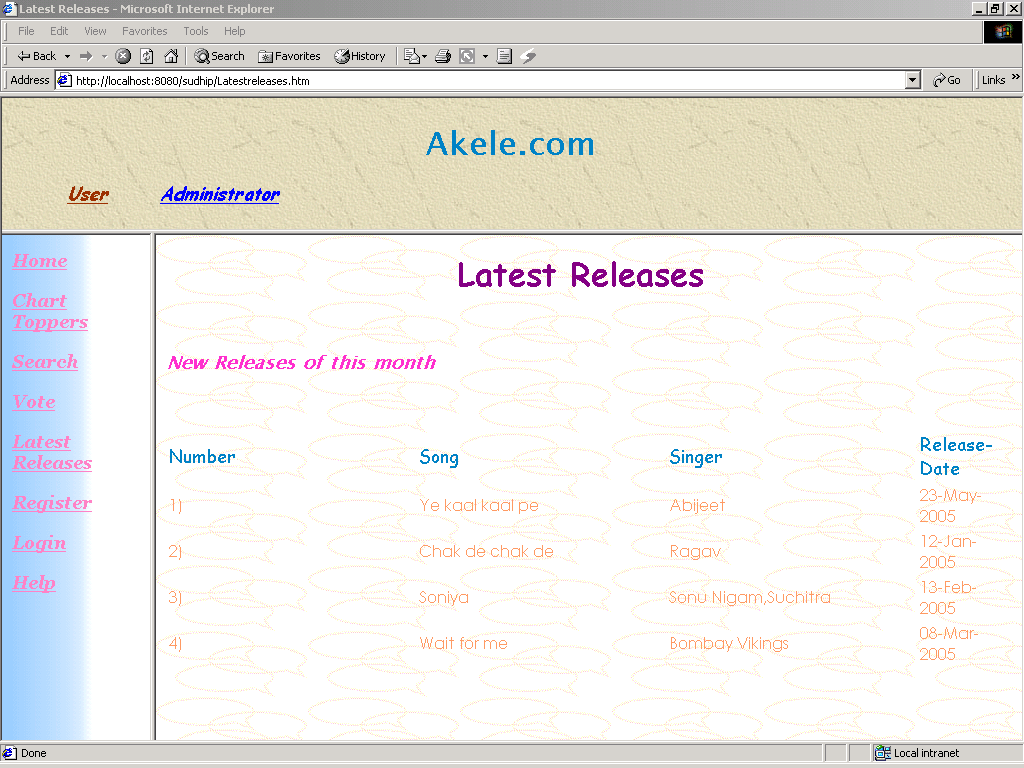
Home Page

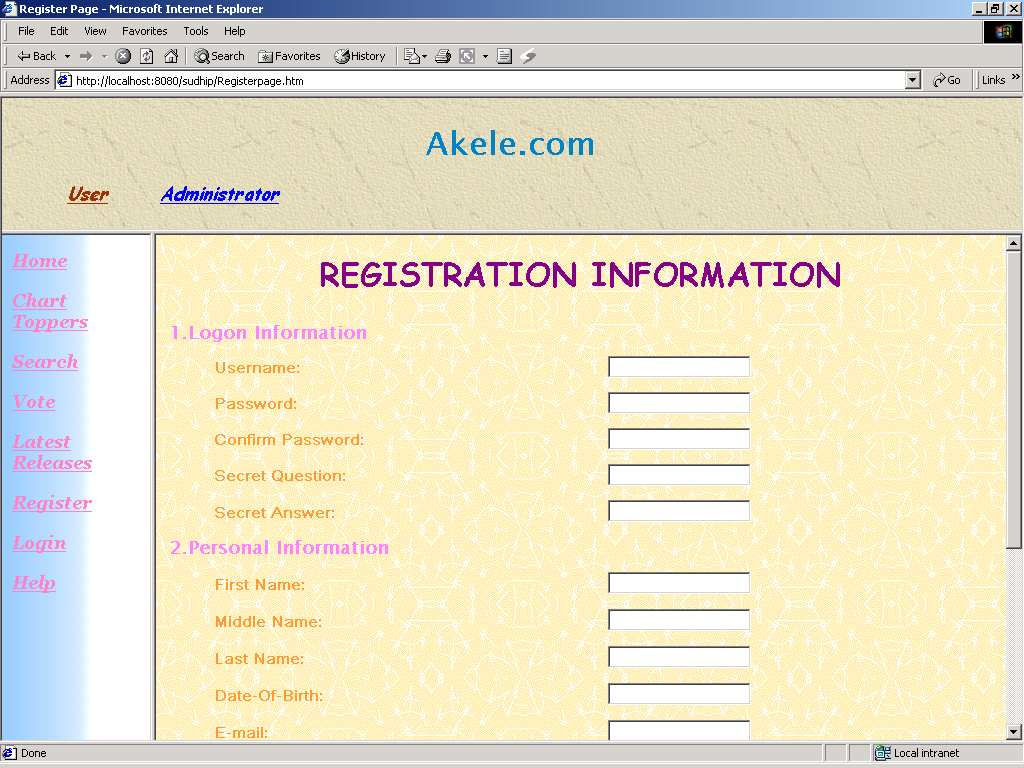


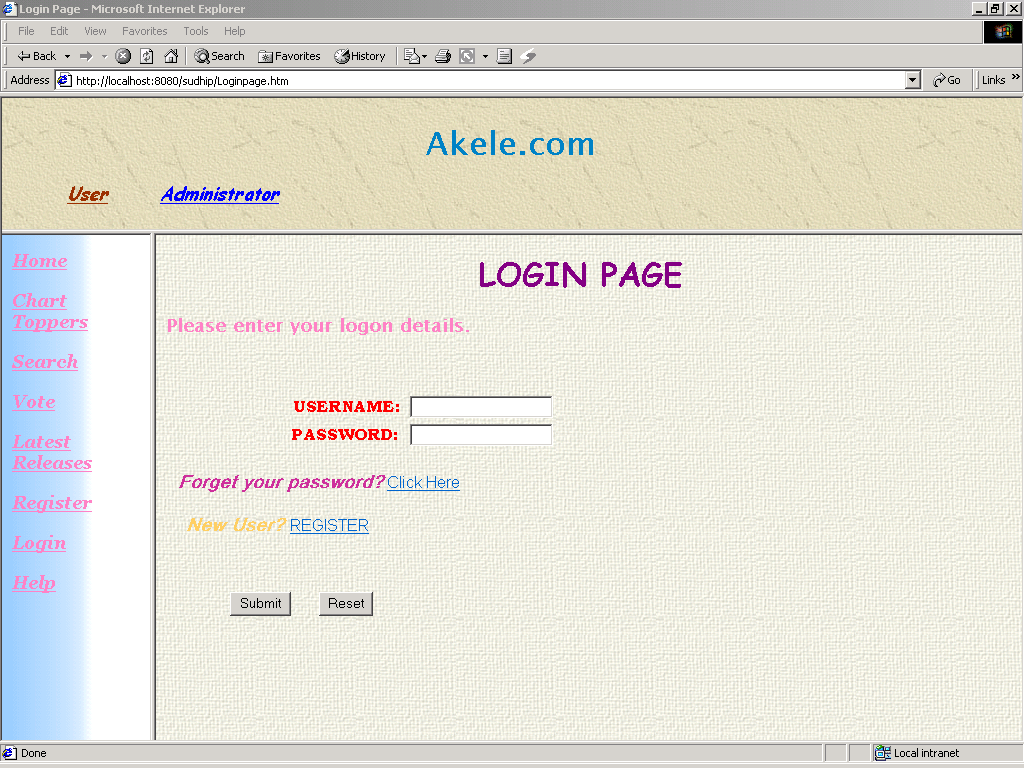


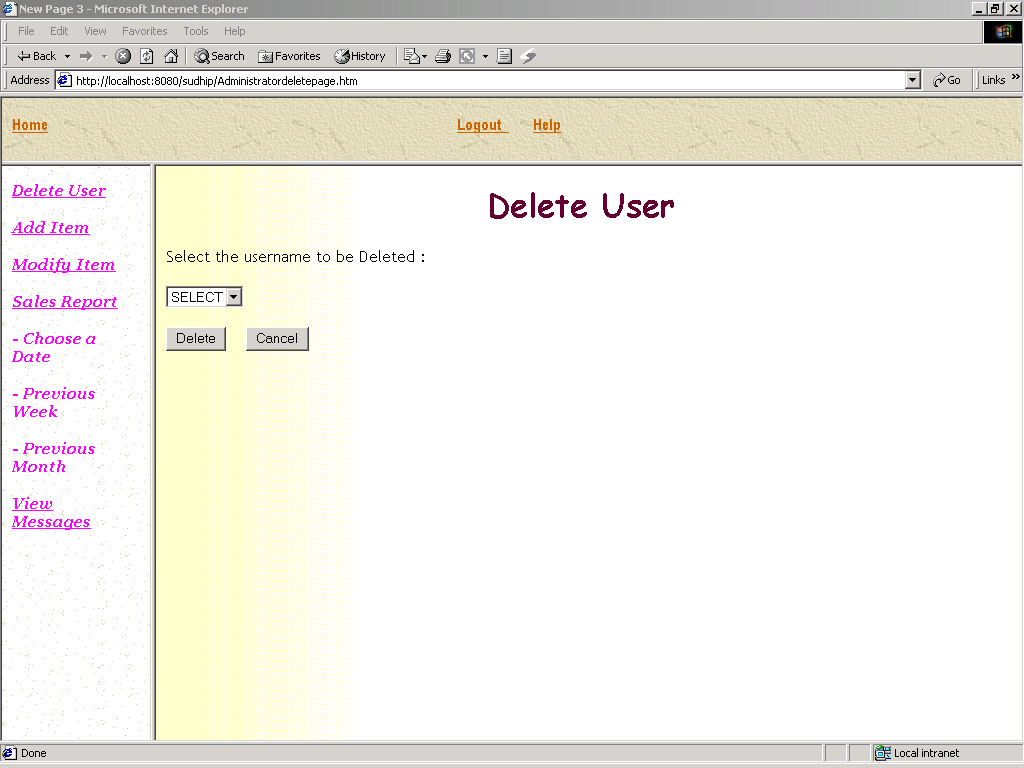


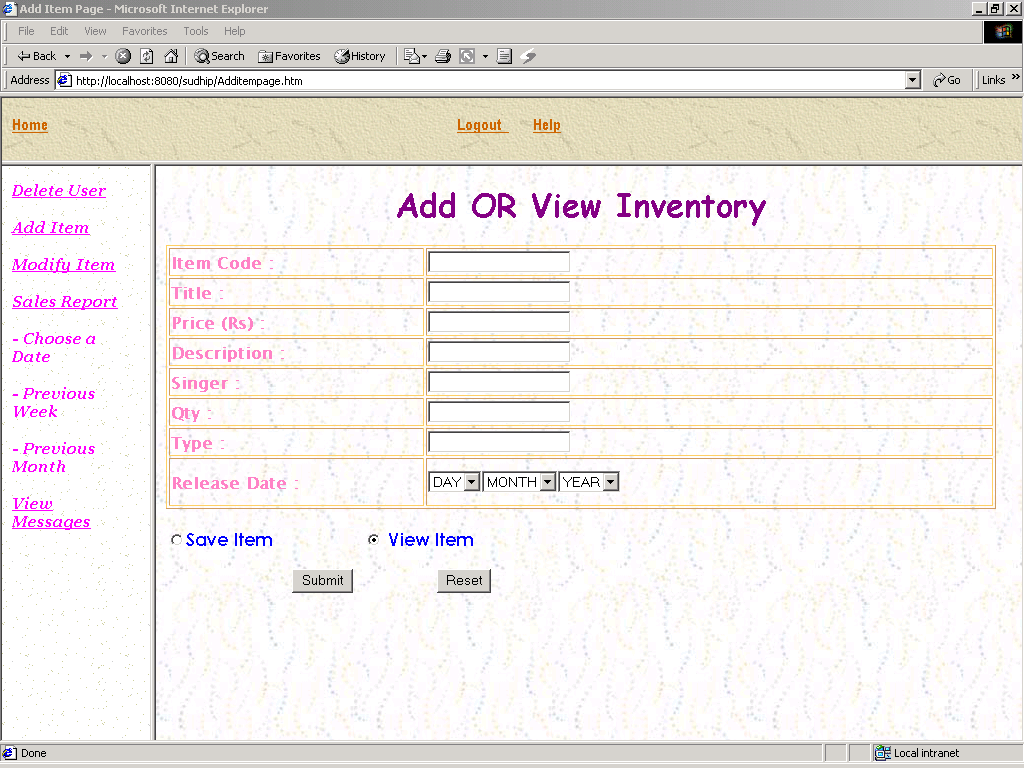


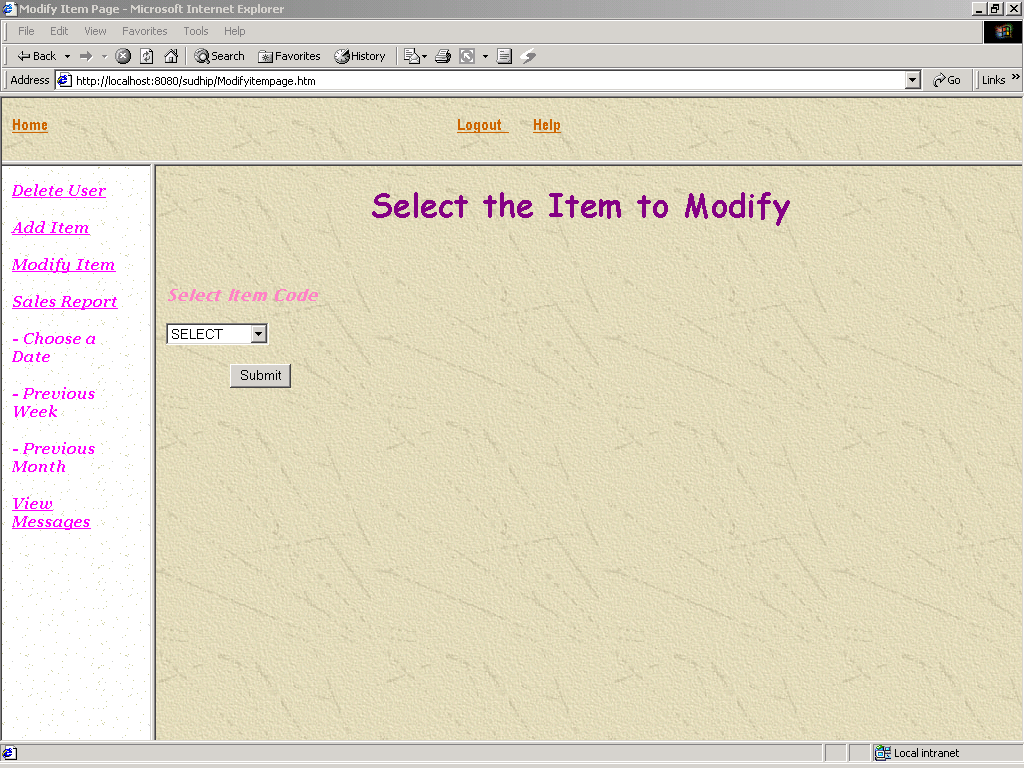


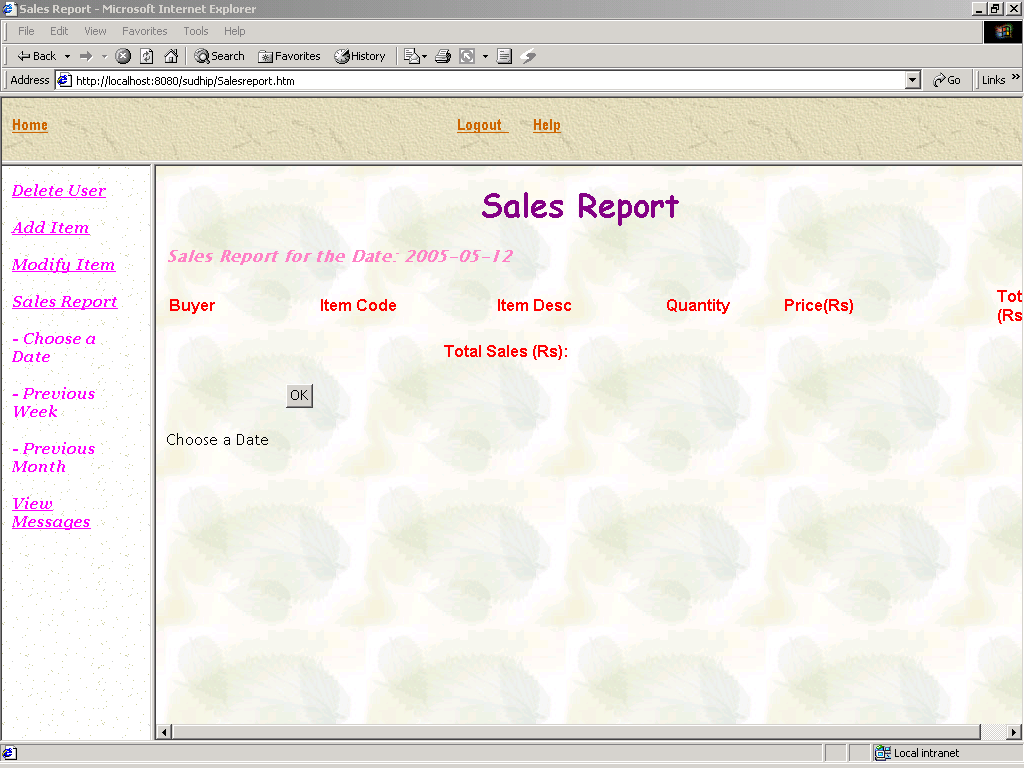


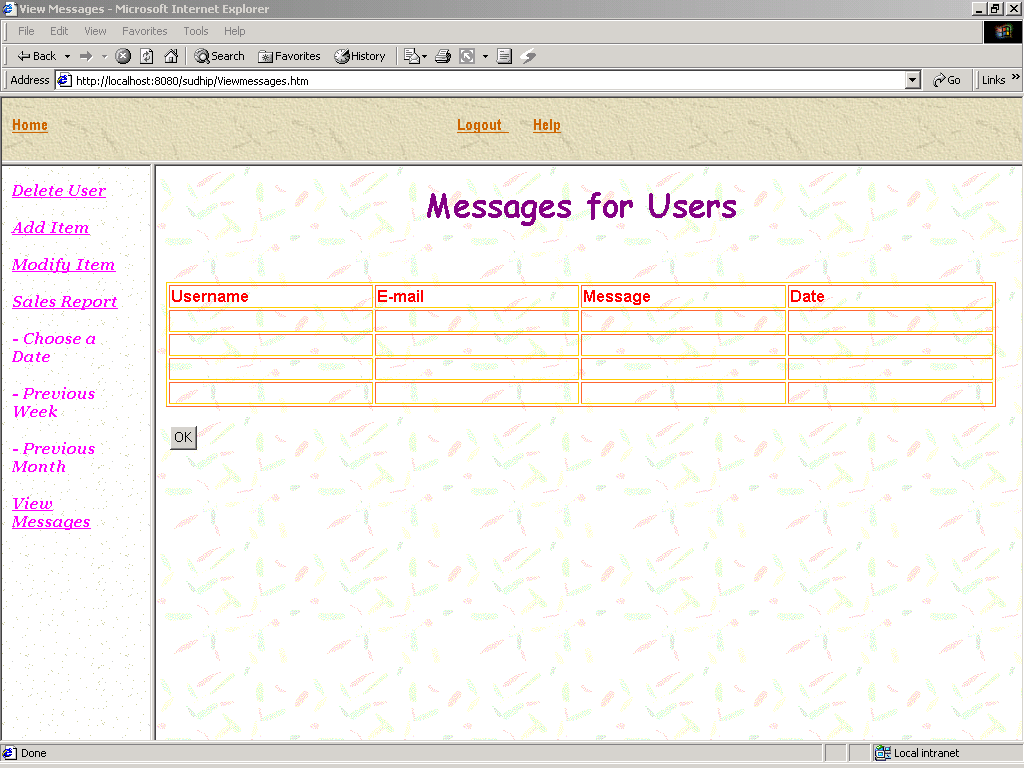


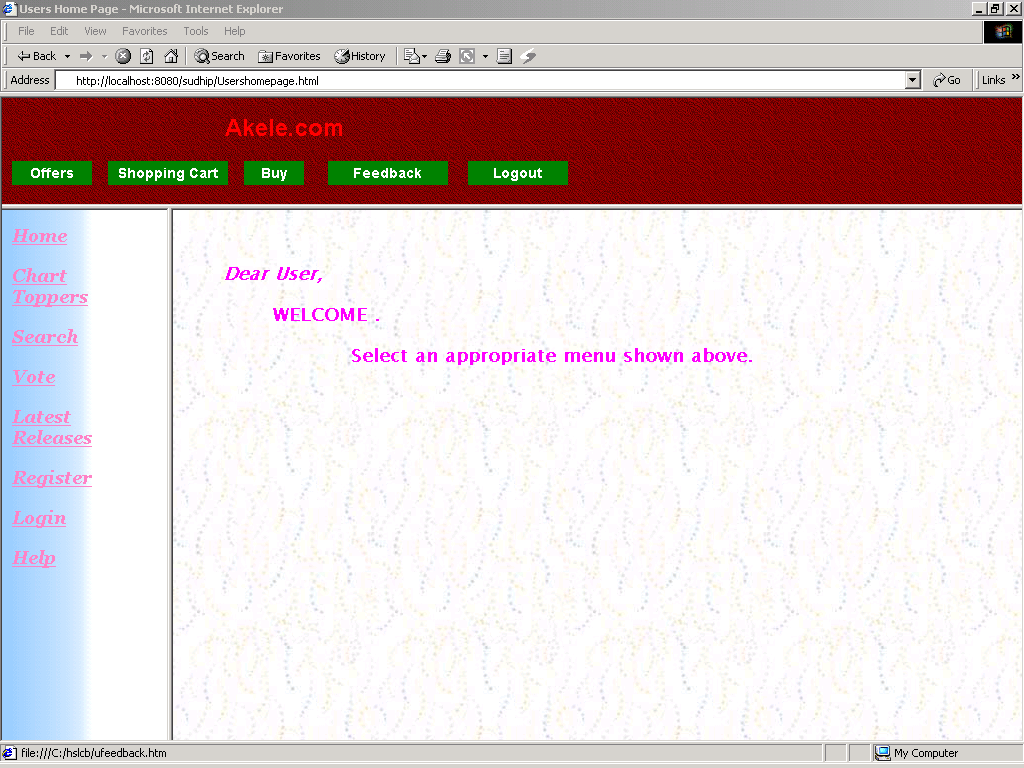


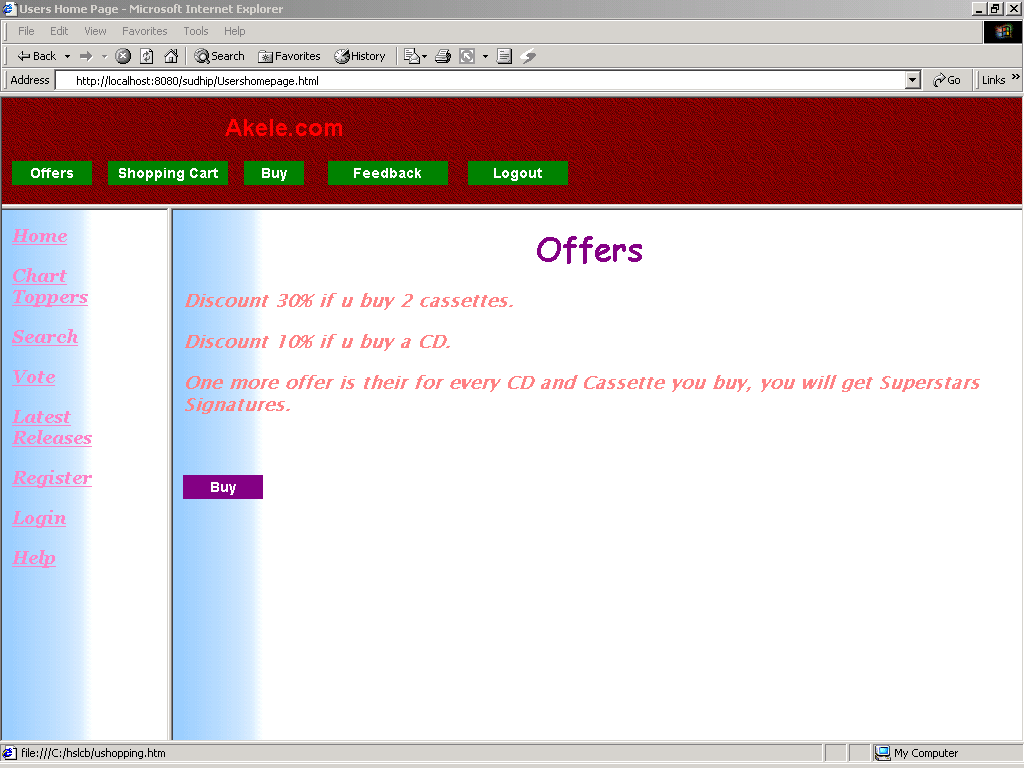


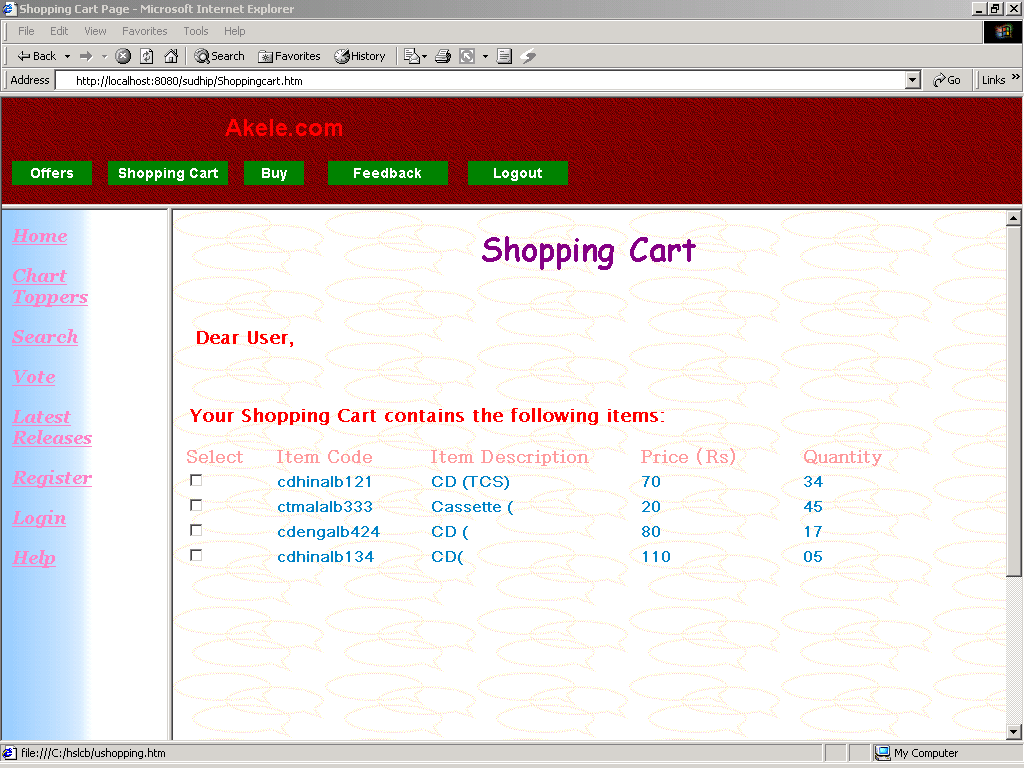


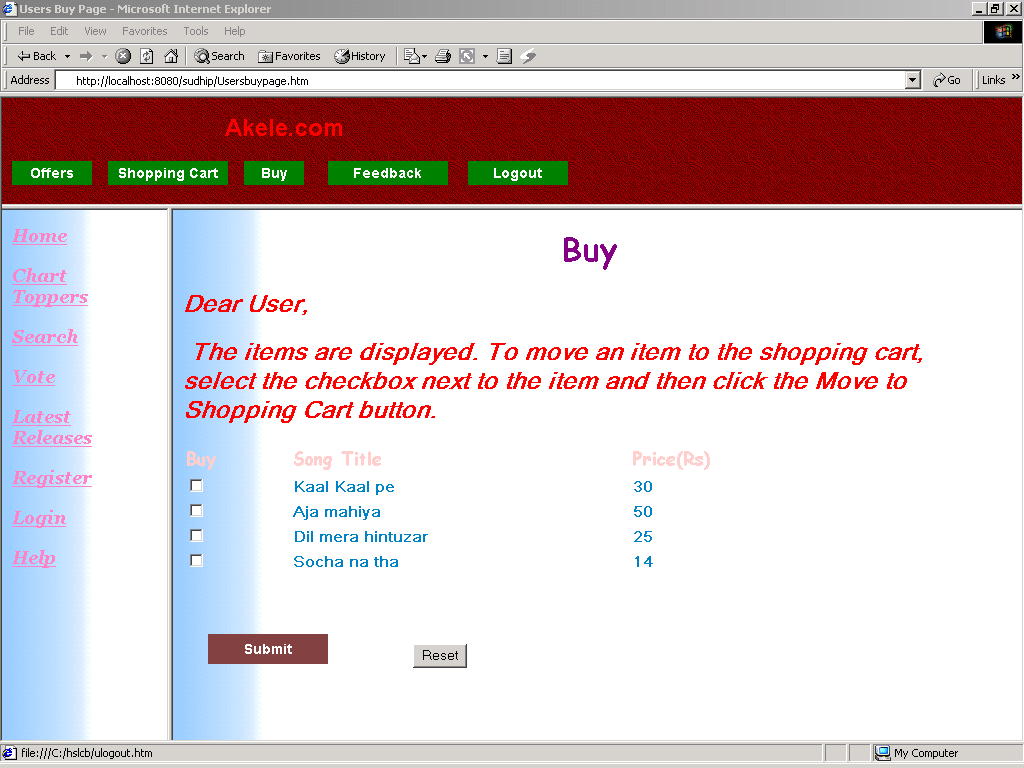


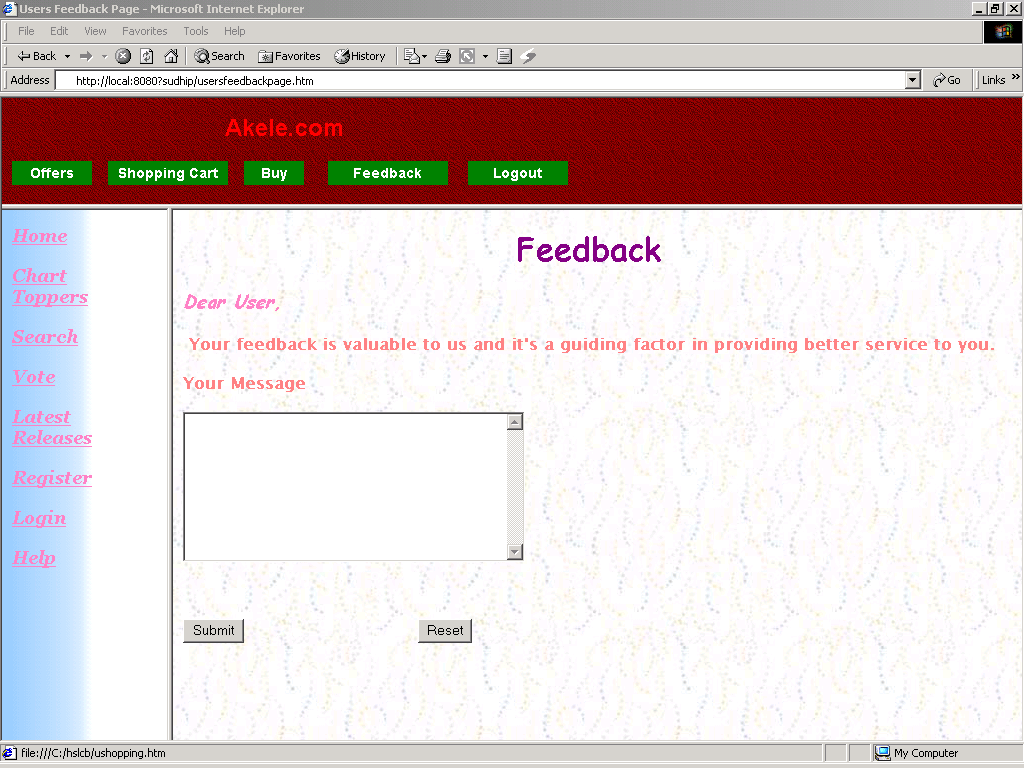


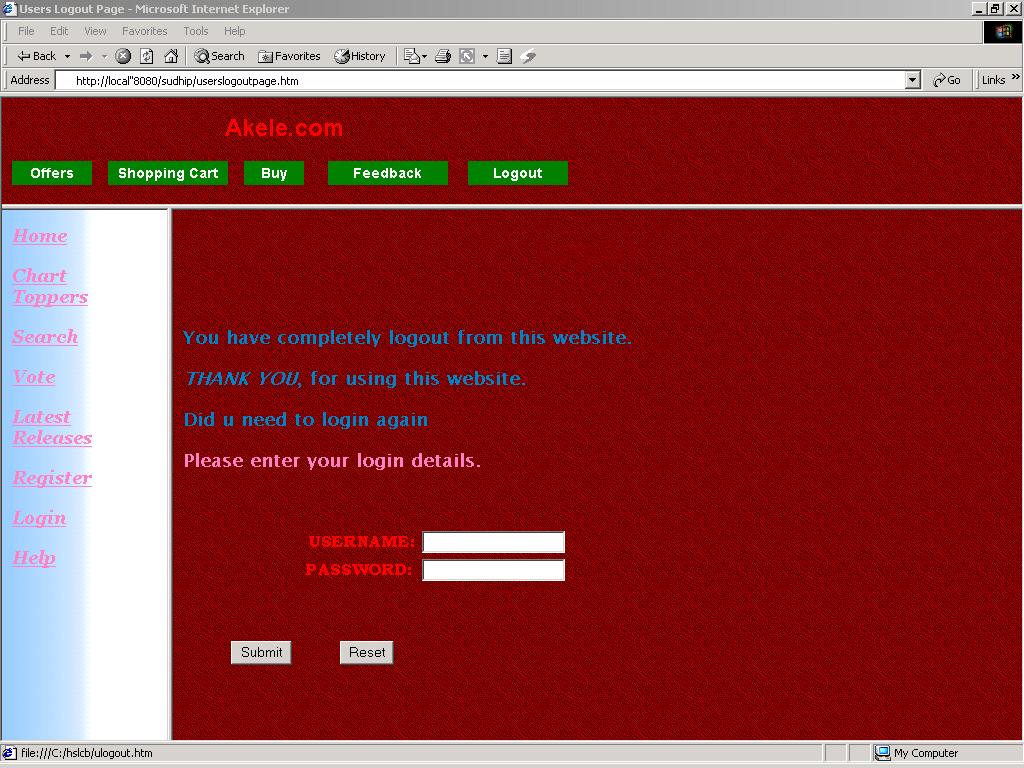












**CODING**

**JSP Code:-**

**about.jsp**

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/amainmenu.jsp" %>

<div id="page">

<%@ include file="menu/sidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<%@ include file="add/ad90.jsp" %>

<br>

<br>

<h2 align='center' >Send Your Feedback here</h2>

<br>

<table align="center">

<tr>

<td align="center"><h2>Music WorlD</h2>

<p>Email Address : songs.fr.ms@gmail.com </p>

<p>&nbsp;</p></td>

</tr>

</table>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**aindex.jsp**

<%@ include file="asession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script language="javascript" type="text/javascript">

function valid()

{var v=true;

if(document.getElementById("name").value=="")

{

alert("Enter Movie name");

v=false;

}

else if(document.getElementById("l128").value=="")

{

alert("Enter Link 128kb");

v=false;

}

else if(document.getElementById("l320").value=="")

{

alert("Enter Link 320kb");

v=false;

}

else if(document.getElementById("limg").value=="")

{

alert("Enter Link img");

v=false;

}

return v;

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/adminsidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Add New Movie Song</h1>

<center>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br>"+msg);

%>

<BR><BR>

<form id="formID" method="get" action="addnew.jsp">

<table width="350px" border="0">

<tr height="60px">

<td width="136" align="center">Name:

</td>

<td width="152"><input type="text" name="name" id="name" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link 128kb:

</td>

<td width="152"><input type="text" name="l128" id="l128" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link 320kb:

</td>

<td width="152"><input type="text" name="l320" id="l320" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link Image:

</td>

<td width="152"><input type="text" name="limg" id="limg" /></td>

</tr>

<tr align="center" height="50px">

<td>

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>Pop</span>

<input type="radio" name="type" id="radio1" value="p">

</td></tr>

<tr height="60px"><td colspan="2" align="center">

<input class="submit" type="submit" value="Submit" onclick="return valid();"/>

</td>

</tr>

</table>

</form>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/afooter.jsp" %>

</div>

</body></html>

**asession.jsp**

<%

String ses[] = session.getValueNames();

if(ses.length==0){

response.sendRedirect("login.jsp");

}

%>

**bolly.jsp**

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/bmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/sidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Download Bollywood Songs Here ..!</h1>

<%@ include file="add/ad90.jsp" %>

<br>

<%@ include file="lastbollyupdate.jsp" %>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**DConnection.jsp**

<%@page import="java.sql.\*" %>

<%!

public class DConnection {

Connection con;

ResultSet rs=null;

Statement st;

public int setData(String str){

int i=0;

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost/jayvik123?user=root&password=jayvik");

st=con.createStatement();

st.execute(str);

i = st.getUpdateCount();

}catch(Exception e){

}

return i;

}

public ResultSet getData(String str) {

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost/jayvik123?user=root&password=jayvik");

st=con.createStatement();

rs = st.executeQuery(str);

}

catch (Exception e){

}

return rs;

}

}

%>

**delete.jsp**

<%@ include file="asession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script type="text/javascript">

function loadXMLDoc()

{

var tex = document.srch.pname.value;

if(tex!=""){

for (var i=0; i<document.srch.type.length; i++)

{

if (document.srch.type[i].checked)

{

var rad\_val = document.srch.type[i].value;

var url = 'dsdata.jsp?value='+tex+'&type='+rad\_val;

}

}

if (window.XMLHttpRequest)

{

xmlhttp=new XMLHttpRequest();

}

else

{

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/adminsidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Delete Movie Song</h1>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br><center>"+msg+"</center><br>");

%>

<form name="srch" id="formID" method="get" action="">

<table width="95%" border="0" align="center" >

<tr height="40px">

<td align="center">

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>Pop</span>

<input type="radio" name="type" id="radio1" value="p">

</td>

<td width="152" align="right"><input onKeyUp="loadXMLDoc()" type="text" name="pname" id="pname" /></td>

<td width="150px" align="left">

<input class="submit" type="button" onClick="loadXMLDoc()" value="search"/>

</td>

</tr>

</table>

</form>

<BR><BR>

<div id="myDiv"></div>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/afooter.jsp" %>

</div>

</body></html>

**deletedata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

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

if(id!=""){

DConnection db = new DConnection();

String qry1="delete from movie where id='"+id+"'";

db.setData(qry1);

qry1="delete from link where id='"+id+"'";

db.setData(qry1);

response.sendRedirect("delete.jsp?msg=delete sucessful");

}

else{

response.sendRedirect("delete.jsp?msg=delete Unsucessful");

}

%>

**download.jsp**

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/sidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Download Bollywood Songs Here ..!</h1>

<%@ include file="add/ad90.jsp" %>

<%

String id="";

id = request.getParameter("id");

if(id!=null){

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select m.name,l.link128,l.link320,l.limg from movie m,link l where m.id=l.id and l.id='"+id+"'";

rs1 = db.getData(qry);

String name="",l128="",l320="",limg="";

while(rs1.next()){

name = rs1.getString("m.name");

l128 = rs1.getString("l.link128");

l320 = rs1.getString("l.link320");

limg = rs1.getString("l.limg");

%>

<br>

<table align='center' border='0'width="84%">

<tr><td align='center'><h1><% out.println(name); %><h1></td><td width='170px' height='170px'><img src='<%=limg %>' width='170px' height='170px'></img></td><tr>

<tr align='center' height='65px'><td>All Songs Zipfile 120Kbps &nbsp;&nbsp;&nbsp;&nbsp;<a href='<%=l128 %>' >Download here</a></td></tr>

<tr align='center' height='65px'><td>All Songs Zipfile 320Kbps &nbsp;&nbsp;&nbsp;&nbsp;<a href='<%=l320 %>' >Download here</a></td></tr>

</table>

<%

}

}

%>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**dsdata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

String id="";

id = request.getParameter("value");

if(id!=null){

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

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select m.id,m.name from movie m,link l where m.id=l.id and m.type='"+type+"' and m.name LIKE '"+id+"%'";

rs1 = db.getData(qry);

String pid="",nm="";

out.println("<table border='0' align='center' width='94%'><tr align='center' height='50px'><td><b>ID</b></td><td><b>Name</b></td><td></td></tr>");

while(rs1.next()){

pid = rs1.getString("m.id");

nm = rs1.getString("m.name");

%>

<tr align='center'>

<td><% out.println(pid); %></td>

<td><% out.println(nm); %></td>

<td><a href='deletedata.jsp?id=<%=pid %>' >Delete</a></td>

</tr>

<% }

out.println("</table>");

}

%>

**index.jsp**

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/mainmenu.jsp" %>

<div id="page">

<%@ include file="menu/isidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Download Bollywood Songs Here ..!</h1>

<%@ include file="add/ad90.jsp" %>

<br>

<%@ include file="lastupdate.jsp" %>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**isession.jsp**

<%

String ses[] = session.getValueNames();

if(ses.length==2){

response.sendRedirect("login.jsp");

}

%>

lastbollyupdate.jsp

<%@page import="java.sql.\*" %><%@include file="DConnection.jsp" %> <% DConnection db = new DConnection(); ResultSet rs1=null; String qry = "select \* from movie where type='b' ORDER BY date DESC LIMIT 15"; rs1 = db.getData(qry); String name="",type="",id=""; out.println("<h1>New Songs</h1><br>"); out.println("<table border='0' align='center' width='94%'><tr><td width='50%'>"); out.println("<table border='0' width='100%'><tr align='center' height='50px'><td><b>Movie Name</b></td><td><b>Type</b></td></tr>"); while(rs1.next()){ id = rs1.getString("id"); name = rs1.getString("name"); type = rs1.getString("type"); %> <tr align='center' > <td><a href='download.jsp?id=<%=id %>' ><% out.println(name); %></a></td> <td><% out.println("Movie"); %></td> </tr> <% } out.println("</table>"); out.println("</td><td></td></tr></table>"); %>

lastpopupupdate.jsp

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select \* from movie where type='p' ORDER BY date DESC LIMIT 15";

rs1 = db.getData(qry);

String name="",type="",id="";

out.println("<h1>New Songs</h1><br>");

out.println("<table border='0' align='center' width='94%'><tr><td width='50%'>");

out.println("<table border='0' width='100%'><tr align='center' height='50px'><td><b>Albums Name</b></td><td><b>Type</b></td></tr>");

while(rs1.next()){

id = rs1.getString("id");

name = rs1.getString("name");

type = rs1.getString("type");

%>

<tr align='center' >

<td><a href='download.jsp?id=<%=id %>' ><% out.println(name); %></a></td>

<td><% out.println("Pop"); %></td>

</tr>

<% }

out.println("</table>");

out.println("</td><td></td></tr></table>");

%>

lastupdate.jsp

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select \* from movie ORDER BY date DESC LIMIT 15";

rs1 = db.getData(qry);

String name="",type="",id="";

out.println("<h1>New Songs</h1><br>");

out.println("<table border='0' align='center' width='94%'><tr><td width='55%'>");

out.println("<table border='0' width='100%'><tr align='center' height='50px'><td><b>Movie Name</b></td><td width='30%'><b>Type</b></td></tr>");

while(rs1.next()){

id = rs1.getString("id");

name = rs1.getString("name");

type = rs1.getString("type");

%>

<tr align='center' >

<td><a href='download.jsp?id=<%=id %>' ><% out.println(name); %></a></td>

<td><%

if(type.equals("b")){

out.println("Movie");

}

else{

out.println("Pop");

}

%></td>

</tr>

<% }

out.println("</table>");

out.println("</td><td></td></tr></table>");

%>

**logcheck.jsp**

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%

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

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

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select \* from login where lid='"+user+"' and pwd ='"+pwd+"'";

rs1 = db.getData(qry);

String u="",p="";

while(rs1.next()){

u = rs1.getString("lid");

p = rs1.getString("pwd");

}

if(u.equals(""))

response.sendRedirect("login.jsp?msg=User Name or PassWord Worng");

else{

session.setAttribute("username",u);

response.sendRedirect("aindex.jsp");

}

%>

**login.jsp**

<%@ include file="isession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/mainmenu.jsp" %>

<div id="page">

<%@ include file="menu/sidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<a name="fluidity"></a>

<h1>Login only for Admin ..!</h1>

<center>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br>"+msg);

%>

<BR><BR>

<form id="formID" method="get" action="logcheck.jsp">

<table width="350px" border="0">

<tr height="60px">

<td width="136" align="center"> User Name:

</td>

<td width="152"><input class="validate[required,custom[noSpecialCaracters],length[0,20],ajax[ajaxUser]]" type="text" name="user" id="user" /></td>

</tr>

<tr height="60px"><td width="136" align="center">PassWord : </td>

<td>

<input class="validate[required,length[2,11]]" type="password" name="password" id="firstname" /> </td>

<tr height="60px"><td colspan="2" align="center">

<input class="submit" type="submit" value="Log In"/>

</td>

</tr>

</table>

</form>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**logout.jsp**

<%

session.removeAttribute("username");

response.sendRedirect("index.jsp");

%>

**pop.jsp**

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/pmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/sidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Download Bollywood Songs Here ..!</h1>

<%@ include file="add/ad90.jsp" %>

<br>

<%@ include file="lastpopupdate.jsp" %>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**search.jsp**

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script type="text/javascript">

function loadXMLDoc()

{

var tex = document.srch.pname.value;

if(tex!=""){

for (var i=0; i<document.srch.type.length; i++)

{

if (document.srch.type[i].checked)

{

var rad\_val = document.srch.type[i].value;

var url = 'srchdata.jsp?value='+tex+'&type='+rad\_val;

}

}

if (window.XMLHttpRequest)

{

xmlhttp=new XMLHttpRequest();

}

else

{

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/smainmenu.jsp" %>

<div id="page">

<%@ include file="menu/sidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Search Movie Song</h1>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br><center>"+msg+"</center><br>");

%>

<form name="srch" id="formID" method="get" action="">

<table width="95%" border="0" align="center" >

<tr height="40px">

<td align="center">

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>Pop</span>

<input type="radio" name="type" id="radio1" value="p">

</td>

<td width="152" align="right"><input onKeyUp="loadXMLDoc()" type="text" name="pname" id="pname" /></td>

<td width="150px" align="left">

<input class="submit" type="button" onClick="loadXMLDoc()" value="search"/>

</td>

</tr>

</table>

</form>

<BR><BR>

<div id="myDiv"></div>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/footer.jsp" %>

</div>

</body></html>

**searchdata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%

String id="";

id = request.getParameter("value");

if(id!=null){

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

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select m.id,m.name,l.link128,l.link320,l.limg from movie m,link l where m.id=l.id and m.type='"+type+"' and m.name LIKE '"+id+"%' ORDER BY m.name";

rs1 = db.getData(qry);

String pid="",nm="",l12="",l32="",lim="";

int i=0;

out.println("<table border='0' align='center' width='94%'><tr align='center' height='50px'><td><b>Movie Name</b></td><td><b>Movie Name</b></td>");

while(rs1.next()){

pid = rs1.getString("m.id");

nm = rs1.getString("m.name");

if(i%2==0){

out.println("</tr><tr>");

}

out.println("<td align='center'><a href='download.jsp?id="+pid+"' >"+nm+"</a></td>");

i++;

}

out.println("</tr>");

out.println("</table>");

}

%>

**update.jsp**

<%@ include file="asession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script type="text/javascript">

function loadXMLDoc()

{

var tex = document.srch.pname.value;

if(tex!=""){

for (var i=0; i<document.srch.type.length; i++)

{

if (document.srch.type[i].checked)

{

var rad\_val = document.srch.type[i].value;

var url = 'usdata.jsp?value='+tex+'&type='+rad\_val;

}

}

if (window.XMLHttpRequest)

{

xmlhttp=new XMLHttpRequest();

}

else

{

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/adminsidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Update Movie Song</h1>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br><center>"+msg+"</center><br>");

%>

<form name="srch" id="formID" method="get" action="">

<table width="95%" border="0" align="center" >

<tr height="40px">

<td align="center">

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>Pop</span>

<input type="radio" name="type" id="radio1" value="p">

</td>

<td width="152" align="right"><input onKeyUp="loadXMLDoc()" type="text" name="pname" id="pname" /></td>

<td width="150px" align="left">

<input class="submit" type="button" onClick="loadXMLDoc()" value="search"/>

</td>

</tr>

</table>

</form>

<BR><BR>

<div id="myDiv"></div>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/afooter.jsp" %>

</div>

</body></html>

**updatedata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

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

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

String l128 = request.getParameter("l128");

String l320 = request.getParameter("l320");

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

if(id!=""&&name!=""&&l128!=""&&l320!=""&&limg!=""){

DConnection db = new DConnection();

String qry1="update movie set name='"+name+"' where id='"+id+"'";

db.setData(qry1);

qry1="update link set link128='"+l128+"',link320='"+l320+"',limg='"+limg+"' where id='"+id+"'";

db.setData(qry1);

response.sendRedirect("update.jsp?msg=Update sucessful");

}

else{

response.sendRedirect("update.jsp?msg=Update Unsucessful");

}

%>

usdata.jsp

<%@page import="java.sql.\*" %>

<%@include file="DConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

String id="";

id = request.getParameter("value");

if(id!=null){

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

DConnection db = new DConnection();

ResultSet rs1=null;

String qry = "select m.id,m.name,l.link128,l.link320,l.limg from movie m,link l where m.id=l.id and m.type='"+type+"' and m.name LIKE '"+id+"%'";

rs1 = db.getData(qry);

String pid="",nm="",l12="",l32="",lim="";

out.println("<table border='0' align='center' width='94%'><tr align='center' height='50px'><td><b>ID</b></td><td><b>Name</b></td><td><b>Link 128</b><td><b>Link 320</b></td><td><b>link img</b></td><td></td></tr>");

while(rs1.next()){

pid = rs1.getString("m.id");

nm = rs1.getString("m.name");

l12 = rs1.getString("l.link128");

l32 = rs1.getString("l.link320");

lim = rs1.getString("l.limg");

%>

<form method="get" action="updatedata.jsp" >

<tr align='center'>

<td><% out.println(pid); %>

<input type='hidden' value=<%=pid %> name='id' />

</td>

<td><input type="text" size="11" value="<%=nm%>" name="name" id="name" /></td>

<td><input type="text" size="11" value="<%=l12%>" name="l128" id="l128" /> </td>

<td><input type="text" size="11" value="<%=l32%>" name="l320" id="l320" /></td>

<td><input type="text" size="11" value="<%=lim%>" name="limg" id="limg" /></td>

<td><input type="submit" value="UPDATE" ></td>

</tr>

</form>

<% }

out.println("</table>");

}

%>

**vaddnew.jsp**

<%@page import="java.sql.\*" %>

<%@include file="VDConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

String name="";

name = request.getParameter("name");

if(name!=null){

String vlmp4 = request.getParameter("vlmp4");

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

String vl3gp = request.getParameter("vl3gp");

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

String type1 = request.getParameter("type");

VDConnection db = new VDConnection();

ResultSet rs1=null;

String qry = "select vname from video where vname='"+name+"'";

rs1 = db.getData(qry);

String pc="";

while(rs1.next()){

pc = rs1.getString("vname");

}

if(pc.equals("")){

String nb = "Select MAX(vid)'No' from video";

rs1 = db.getData(nb);

int id=0;

while(rs1.next()){

id = rs1.getInt("No")+1;

}

String qry1 = "insert into video values('"+id+"','"+name+"','"+type1+"','"+vlmp4+"','"+vlavi+"','"+vl3gp+"','"+vlimg+"',CURDATE()+0)";

db.setData(qry1);

response.sendRedirect("vaindex.jsp?msg=Video Add Sucessfuly");

}

else{

response.sendRedirect("vaindex.jsp?msg=Video Already In Database");

}

}

%>

**vaindex.jsp**

<%@ include file="asession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script language="javascript" type="text/javascript">

function valid()

{var v=true;

if(document.getElementById("name").value=="")

{

alert("Enter Movie name");

v=false;

}

else if(document.getElementById("vlmp4").value=="")

{

alert("Enter Link Mp4");

v=false;

}

else if(document.getElementById("vlavi").value=="")

{

alert("Enter Link avi");

v=false;

}

else if(document.getElementById("vl3gp").value=="")

{

alert("Enter Link 3gp");

v=false;

}

else if(document.getElementById("vlimg").value=="")

{

alert("Enter Link image");

v=false;

}

return v;

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/adminsidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Add New Video Song</h1>

<center>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br>"+msg);

%>

<BR><BR>

<form id="formID" method="get" action="vaddnew.jsp">

<table width="350px" border="0">

<tr height="60px">

<td width="136" align="center">Name:

</td>

<td width="152"><input type="text" name="name" id="name" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link Mp4:

</td>

<td width="152"><input type="text" name="vlmp4" id="vlmp4" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link avi:

</td>

<td width="152"><input type="text" name="vlavi" id="vlavi" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link 3gp:

</td>

<td width="152"><input type="text" name="vl3gp" id="vl3gp" /></td>

</tr>

<tr height="60px">

<td width="136" align="center">Link image:

</td>

<td width="152"><input type="text" name="vlimg" id="vlimg" /></td>

</tr>

<tr align="center" height="50px">

<td>

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>English</span>

<input type="radio" name="type" id="radio1" value="e">

</td></tr>

<tr height="60px"><td colspan="2" align="center">

<input class="submit" type="submit" value="Submit" onclick="return valid();"/>

</td>

</tr>

</table>

</form>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/afooter.jsp" %>

</div>

</body></html>

**VDConnection.jsp**

<%@page import="java.sql.\*" %>

<%!

public class VDConnection {

Connection con;

ResultSet rs=null;

Statement st;

public int setData(String str){

int i=0;

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost/jayvik12?user=root&password=jayvik");

st=con.createStatement();

st.execute(str);

i = st.getUpdateCount();

}catch(Exception e){

}

return i;

}

public ResultSet getData(String str) {

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost/jayvik12?user=root&password=jayvik");

st=con.createStatement();

rs = st.executeQuery(str);

}

catch (Exception e){

}

return rs;

}

}

%>

**vdelete.jsp**

<%@ include file="asession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script type="text/javascript">

function loadXMLDoc()

{

var tex = document.srch.pname.value;

if(tex!=""){

for (var i=0; i<document.srch.type.length; i++)

{

if (document.srch.type[i].checked)

{

var rad\_val = document.srch.type[i].value;

var url = 'vdsdata.jsp?value='+tex+'&type='+rad\_val;

}

}

if (window.XMLHttpRequest)

{

xmlhttp=new XMLHttpRequest();

}

else

{

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/adminsidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Delete Video Song</h1>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br><center>"+msg+"</center><br>");

%>

<form name="srch" id="formID" method="get" action="">

<table width="95%" border="0" align="center" >

<tr height="40px">

<td align="center">

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>English</span>

<input type="radio" name="type" id="radio1" value="e">

</td>

<td width="152" align="right"><input onKeyUp="loadXMLDoc()" type="text" name="pname" id="pname" /></td>

<td width="150px" align="left">

<input class="submit" type="button" onClick="loadXMLDoc()" value="search"/>

</td>

</tr>

</table>

</form>

<BR><BR>

<div id="myDiv"></div>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/afooter.jsp" %>

</div>

</body></html>

**vdeletedata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="VDConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

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

if(id!=""){

VDConnection db = new VDConnection();

String qry1="delete from video where vid='"+id+"'";

db.setData(qry1);

response.sendRedirect("vdelete.jsp?msg=delete sucessful");

}

else{

response.sendRedirect("vdelete.jsp?msg=delete Unsucessful");

}

%>

**vdsdata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="VDConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

String id="";

id = request.getParameter("value");

if(id!=null){

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

VDConnection db = new VDConnection();

ResultSet rs1=null;

String qry = "select vid,vname from video where vtype='"+type+"' and vname LIKE '"+id+"%'";

rs1 = db.getData(qry);

String pid="",nm="";

out.println("<table border='0' align='center' width='94%'><tr align='center' height='50px'><td><b>ID</b></td><td><b>Name</b></td><td></td></tr>");

while(rs1.next()){

pid = rs1.getString("vid");

nm = rs1.getString("vname");

%>

<tr align='center'>

<td><% out.println(pid); %></td>

<td><% out.println(nm); %></td>

<td><a href='vdeletedata.jsp?id=<%=pid %>' >Delete</a></td>

</tr>

<% }

out.println("</table>");

}

%>

**vupdate.jsp**

<%@ include file="asession.jsp" %>

<html><head>

<title>Download Bollywood Songs</title>

<link media="screen, projection, tv " href="css/html.css" type="text/css" rel="stylesheet">

<link media="screen, projection, tv" href="css/layout.css" type="text/css" rel="stylesheet">

<script type="text/javascript">

function loadXMLDoc()

{

var tex = document.srch.pname.value;

if(tex!=""){

for (var i=0; i<document.srch.type.length; i++)

{

if (document.srch.type[i].checked)

{

var rad\_val = document.srch.type[i].value;

var url = 'vusdata.jsp?value='+tex+'&type='+rad\_val;

}

}

if (window.XMLHttpRequest)

{

xmlhttp=new XMLHttpRequest();

}

else

{

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

}

</script>

</head><body>

<div id="content">

<%@ include file="menu/header.jsp" %>

<%@ include file="menu/dmainmenu.jsp" %>

<div id="page">

<%@ include file="menu/adminsidemenu.jsp" %>

<div class="width75 floatRight">

<div class="gradient">

<h1>Update Video Song</h1>

<%

String msg="";

msg = request.getParameter("msg");

if(msg!=null)

out.println("<br><center>"+msg+"</center><br>");

%>

<form name="srch" id="formID" method="get" action="">

<table width="95%" border="0" align="center" >

<tr height="40px">

<td align="center">

<span>Bollywood </span>

<input type="radio" name="type" id="radio2" value="b" checked />

</td>

<td>

<span>English</span>

<input type="radio" name="type" id="radio1" value="e">

</td>

<td width="152" align="right"><input onKeyUp="loadXMLDoc()" type="text" name="pname" id="pname" /></td>

<td width="150px" align="left">

<input class="submit" type="button" onClick="loadXMLDoc()" value="search"/>

</td>

</tr>

</table>

</form>

<BR><BR>

<div id="myDiv"></div>

<br>

</center>

</div>

</div>

</div>

</div>

<%@ include file="menu/afooter.jsp" %>

</div>

</body></html>

**vupdatedata.jsp**

<%@page import="java.sql.\*" %>

<%@include file="VDConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

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

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

String vlmp4 = request.getParameter("vlmp4");

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

String vl3gp = request.getParameter("vl3gp");

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

if(id!=""&&vname!=""&&vlmp4!=""&&vlavi!=""&&vl3gp!=""&&vlimg!=""){

VDConnection db = new VDConnection();

String qry1="update video set vname='"+vname+"',vlmp4='"+vlmp4+"',vlavi='"+vlavi+"',vl3gp='"+vl3gp+"',vlimg='"+vlimg+"' where vid='"+id+"'";

db.setData(qry1);

response.sendRedirect("vupdate.jsp?msg=Update sucessful");

}

else{

response.sendRedirect("vupdate.jsp?msg=Update Unsucessful");

}

%>

**vusdats.jsp**

<%@page import="java.sql.\*" %>

<%@include file="VDConnection.jsp" %>

<%@ include file="asession.jsp" %>

<%

String id="";

id = request.getParameter("value");

if(id!=null){

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

VDConnection db = new VDConnection();

ResultSet rs1=null;

String qry = "select \* from video where vtype='"+type+"' and vname LIKE '"+id+"%'";

rs1 = db.getData(qry);

String pid="",vname="",vlmp4="",vlavi="",vl3gp="",vlimg="";

out.println("<table border='0' align='center' width='94%'><tr align='center' height='50px'><td><b>ID</b></td><td><b>Name</b></td><td><b>Mp4</b><td><b>avi</b></td><td><b>3gp</b></td><td><b>image</b></td><td></td></tr>");

while(rs1.next()){

pid = rs1.getString("vid");

vname = rs1.getString("vname");

vlmp4 = rs1.getString("vlmp4");

vlavi = rs1.getString("vlavi");

vl3gp = rs1.getString("vl3gp");

vlimg = rs1.getString("vlimg");

%>

<form method="get" action="vupdatedata.jsp" >

<tr align='center'>

<td><% out.println(pid); %>

<input type='hidden' value=<%=pid %> name='id' />

</td>

<td><input type="text" size="11" value="<%=vname%>" name="vname" id="vname" /></td>

<td><input type="text" size="11" value="<%=vlmp4%>" name="vlmp4" id="vlmp4" /> </td>

<td><input type="text" size="11" value="<%=vlavi%>" name="vlavi" id="vlavi" /></td>

<td><input type="text" size="11" value="<%=vl3gp%>" name="vl3gp" id="vl3gp" /></td>

<td><input type="text" size="11" value="<%=vlimg%>" name="vlimg" id="vlimg" /></td>

<td><input type="submit" value="UPDATE" ></td>

</tr>

</form>

<% }

out.println("</table>");

}

%>

**Cascading Style Sheet:-**

**html.css**

html {

height: 100%;

}

body {

height: 100%;

margin: 0;

padding: 0;

text-align: center;

font: 400 0.8em verdana, arial, sans-serif;

line-height: 170%;

background: #666 url(../images/bg/header.jpg) repeat-x top left;

color: #EEE;

}

h1 {

clear: both;

font: 700 2.5em "trebuchet ms", serif;

color: #FFF;

}

h2 {

font: 400 1.5em "trebuchet ms", serif;

color: #9FF3FF;

}

h2 a {

color: #29E3FF;

}

h2 a:hover {

color: #FFF;

}

h3 {

font: 400 1.3em "trebuchet ms", serif;

color: #F8D766;

}

h4 {

font: 700 1.2em "trebuchet ms", serif;

color: #F39F01;

}

h5 {

font: 700 1.1em "trebuchet ms", serif;

color: #F17400;

}

h6 {

font: 700 1em "trebuchet ms", serif;

color: #FF4B33;

}

h1, h2, h3, h4, h5, h6, p, dl {

margin: 0;

padding: 10px 15px;

}

.leftColumn h1,

.leftColumn h2,

.leftColumn h3,

.leftColumn h4,

.leftColumn h5,

.leftColumn h6,

.leftColumn p,

.leftColumn dl {

padding-left: 0;

}

a {

color: #8BE6FA;

}

a:hover {

color: #FFF;

}

b {

color: #FFF;

}

ol, ul {

margin: 10px 30px;

padding: 0 30px;

}

ol {

color: #FC0;

}

ol span {

color: #EEE;

}

ul {

list-style-image: url(../images/bg/bullet.gif);

}

del {

color: #AAA;

}

code {

margin: 10px 15px;

padding: 10px;

display: block;

overflow: auto;

font: 400 1em courier, monospace;

line-height: 120%;

white-space: pre;

background: #444;

}

.leftColumn code {

margin-left: 0;

}

acronym {

cursor: help;

border-bottom: 1px solid #777;

}

dt {

font-weight: bold;

color: #FFB323;

}

dd {

margin-left: 0;

padding-left: 45px;

}

label {

display: block;

}

input,

textarea,

select {

padding: 2px;

font: 400 1em verdana, sans-serif;

color: #444;

background: #EEE;

border: 1px solid #444;

}

input:focus,

input:hover,

textarea:focus,

textarea:hover,

select:focus,

select:hover {

color: #000;

background: #E4F7FA;

border: 1px solid #00DFFF;

}

input.button {

padding: 2px 5px;

font: 400 1.1em "trebuchet ms", serif;

color: #555;

background: #9FF3FF;

border-width: 1px;

border-style: solid;

border-color: #FFF #00DFFF #00DFFF #FFF;

}

blockquote {

margin: 10px 15px;

padding-left: 27px;

background-color: #444;

background-image: url(../images/icons/quote.gif);

background-repeat: no-repeat;

background-position: 5px 50%;

}

blockquote.exclamation {

background-image: url(../images/icons/exclamation.gif);

}

blockquote.stop {

background-image: url(../images/icons/stop.gif);

}

blockquote.go {

background-image: url(../images/icons/go.gif);

}

img.floatRight {

margin: 5px 0px 10px 15px;

}

img.floatLeft {

margin: 5px 15px 10px 0px;

}

a img {

border: 2px solid #8BE6FA;

}

a:hover img {

border: 2px solid #FFF !important;

border: 2px solid #8BE6FA;

}

**layout.css**

#content {

height: 100%;

min-height: 100%;

text-align: left;

}

#content,

#width {

width: expression(document.body.clientWidth > 1000 ? "1000px" : "94%");

max-width: 1000px;

margin: 0 auto;

}

#content[id],

#width[id] {

width: 94%;

height: auto;

}

#header {

position: relative;

height: 193px;

}

#header #title {

position: absolute;

z-index: 3;

top: 10px;

left: 130px;

padding: 5px;

text-align: right;

}

#header h1 {

margin: 0;

padding: 0;

font: 700 4em "trebuchet ms", serif;

letter-spacing: -3px;

text-transform: lowercase;

color: #FFF;

}

#header h2 {

position: absolute;

top: 10px;

right: 5px;

margin: 0;

padding: 0;

font: 700 1em "trebuchet ms", serif;

text-transform: lowercase;

color: #00F0EC;

}

#header img.left {

position: absolute;

z-index: 1;

top: 0;

left: 0;

}

#header img.right {

position: absolute;

z-index: 0;

top: 0;

right: 0;

}

#header img.balloons {

position: absolute;

z-index: 2;

top: 70px;

right: 400px;

}

#mainMenu {

float: left;

width: 100%;

clear: both;

}

#mainMenu ul {

margin: 0;

padding: 0;

}

#mainMenu li {

display: inline;

list-style: none;

margin: 0;

padding: 0;

}

#mainMenu li a {

float: left;

margin: 0 2px;

padding: 5px 0.5em;

font: 400 1.6em "trebuchet ms", serif;

text-decoration: none;

text-transform: lowercase;

color: #FFF;

}

#mainMenu li a:hover,

#mainMenu li a.here {

color: #65EBFF;

border-top: 5px solid #000;

}

#mainMenu li a.last {

margin-right: 0;

}

ul.sideMenu {

margin: 0;

padding: 0;

}

.sideMenu li {

display: inline;

list-style-image: url(foo.gif);

font: 400 1.3em "trebuchet ms", serif;

}

.sideMenu li a {

display: block;

margin: 0.2em 0;

padding: 3px 5px;

text-decoration: none;

color: #FFF;

}

.sideMenu li a:hover {

color: #65EBFF;

background: #5F5F5F;

}

.sideMenu li.here {

display: block;

padding: 5px;

color: #65EBFF;

background: #555;

}

.sideMenu li.here ul {

margin: 0;

padding: 0;

}

.sideMenu li.here ul li a {

padding-left: 35px;

font: 400 0.55em verdana, arial, sans-serif;

color: #FFF;

background: url(../images/bg/bullet.gif) no-repeat 10px 0px;

}

.sideMenu li.here ul li a:hover {

color: #9FF3FF;

background: #5F5F5F url(../images/bg/bullet.gif) no-repeat 10px 0px;

}

#page {

float: left;

width: 100%;

clear: both;

padding-bottom: 4em;

}

#footer {

float: left;

width: 100%;

clear: both;

margin-top: -3.8em;

background: #000 url(../images/bg/footer.jpg) repeat-x top left;

}

#footer #width {

position: relative;

z-index: 3;

font-size: 0.85em;

padding-top: 27px;

}

.width100 {

width: 100%;

}

.width75 {

width: 74%;

}

.width50 {

width: 49.7%;

}

.width33 {

width: 32.7%;

}

.width25 {

width: 24.7%;

}

.floatLeft {

float: left;

}

.floatRight {

float: right;

}

.alignLeft {

text-align: left;

}

.alignRight {

text-align: right;

}

.clear {

clear: both;

}

.block {

display: block;

}

.small {

font-size: 0.8em;

}

.green {

color: #A1FF45;

}

.red {

color: #EA1B00;

}

.grey {

color: #666;

}

.grey a {

color: #999;

}

.grey a:hover {

color: #EEE;

}

.gradient {

margin-bottom: 2em;

background: #555 url(../images/bg/gradient.jpg) repeat-x bottom left;

}

**CODE EFFICIENCY**

In an effort to maintain computational efficiency and to allow the eventual adaptation of the algorithm to face tracking applications, intense optimization of the code has been performed. Although further development is in progress, the algorithm is currently fast and compact enough to run interactively on most generic platforms.

Note, first, the sequential hierarchical search which proceeds from large scales to small scales. This allows a rapid convergence if the face is dominant in the image. Furthermore, the algorithm does not always flow through the complete loop. It stops as soon as one of the modules reports a failure and loops back to an earlier stage. For example, we do not search for a mouth if no eyes are found. In this case, no time is wasted in the mouth module.

Additionally, in this Project I have utilized special programming techniques to reduce the run-time. The **3D-normalization algorithm** is also extremely efficient and uses look-up tables and minimal calculations for increased speed. The 10 normalization’s and DFFS calculations required for nose-localization also utilize small mug-shot images and increase efficiency.

**OPTIMIZATION OF CODE**

**In my Project** **“MUSIC STORE MANAGEMENT”** I have proposed a method to enable aggressive, inter procedural optimization in a setting where code can be replaced at runtime. Code replacement involves both introducing a new module into the system and reallocating old code. Code purging reallocates replaced code, which is required in long running systems.

My approach, module merging, is simple and practical:

In this Project, I have merged code modules and insert code to check for code replacement at the appropriate points. We show how to preserve the behavior of code purging. The net result is that merged modules preserve the original code replacement behavior, while enabling optimization across code replacement boundaries.

**VALIDATION CHECKS**

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software test - Validation Testing - may begin. Validation can be defined in many ways, but a simple is that validation succeeds when software functions in a manner that can be reasonably expected by the customer.

Validation is an important characteristic of a good project because many times it has been seen that running project may come to a halt stage or just because of ignorance of validation invalid or insufficient data may get enter in the database.

**In this project I adopted many level of validations** like in any entry form user is not supposed to leave any field blank.

As there is a form for customer details there are some fields which are mandatory like name, address, phone no, email id etc. snap of customer (optional) and also some fields are optional because it might be possible that customer does not have any middle name mean to say any unessential field which is not required in data storage.

Then in the **second level of validation** I have checked individual fields like the Name can not be numeric, Phone number should not contain any alphabet, if by mistake any alphabets are entered then the message appears that phone number should be in Digits and in the same context I applied the third level of validation that the phone number should be between **7 to 10 digits** and if user entered data in the number form but not between in this range then the message splash that the number should between 7 to 10 digits.

The password field it is checked that the password shall not be less than four digits. Similarly the Age field, it has been checked that it should not contain any alphabets. Other fields have also been validated which can be found out in the coding.

**TESTING TECHNIQUES AND STRATEGIES**

The philosophy behind the testing is to find errors. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as normal input. There are two general strategies for testing Software: Code Testing and Specification Testing. In code Testing, The analyst develops the cases to execute every instructions and path in the program. Under specification testing, the analyst examines the program specification and then writes test data to determine how the program operates under specific conditions.

**In this Project I have conducted the following Levels of Tests**

# LEVELS OF TESTS

**UNIT TESTING**

In unit testing I have tested the programs making up a system. For this reason, unit testing is sometimes called program testing. Unit Testing gives stress on the modules independently of one another, to find errors. The **MUSIC STORE MANAGEMENT**  consists of modules to handle registration, modify or retrieve data and to respond to different types of inquiries or prints reports. The test cases needed for unit testing should exercise each condition and option.

**SYSTEM TESTING**

This is one of the important and essential part in this Project. Moreover it is an important and essential part of the system development phase, after designing and developing the software is system testing. It can not be said that every program or system design is perfect and because of lack of communication between the user and the designer, some error is there in the software development.

**In this Project I have conducted the following steps of System Testing:**

* String Testing
* Program Testing
* System Testing
* System Documentation
* User Acceptance testing.

## SPECIAL SYSTEM TESTS

There are other **6 tests** that fall under special category. They are:

**PEAK LOAD TEST**

In this project it determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand.

**STORAGE TESTING**

In this project it determines the capacity of the system to store transaction data on a disk or in other files.

**PERFORMANCE TIME TESTING**

In this project it determines the length of time system used by the system to process transaction data.

**RECOVERY TESTING**

In this project this testing determines the ability of user to recover data or restart system after failure.

**PROCEDURE TESTING**

In this project it determines the clarity of documentation on operation and use of system by having users do exactly what manuals request.

**HUMAN FACTORS TESTING**

In this project it determines how the users will use the system when processing data or preparing reports.

**IMPLEMENTATION**

One of the crucial phase in the Software Development Life Cycle is the successful implementation of the software. Implementation includes all those activities that take place to convert from the old system to the new one.

The new system may be completely new, replacing an existing manual or automated system or it may be major modification to an existing system. In either case, proper implementation becomes necessary so that a reliable system based on the requirements of the organization can be provided.

Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it. It has been observed that even the best system cannot show good result if the analysts managing the implementation do not attend to every important detail. This is an area where the systems analysts need to work with utmost care.

In this Project the implementation discusses the three aspects of implementation

* Training personnel
* Conversion procedures
* Post- implementation review

EVALUATION

In this Project much of the management is decision making, according to one of many approaches to management. While there are several views of what constitutes management, according to the decision oriented view, management mainly comprises the following in view of this poject:

* Planning
* Organizing
* Directing and
* Control

Each one of these functions may be at the strategic, tactical or operational level. To illustrate this point we will use a series of examples of strategic, tactical and operational decisions and the information needs, in each of these functional areas.

**PLANNING**

Strategic level planning would call for a lot of environmental information like shifting markets, changing technology as well as internal information like core-competitive strength of the organisation.

Tactical planning activities like vendor development make-or-buy decisions would call for cost and availability information pertaining to materials, production capacities both internal to the organisation as will as outside. Operational planning like staff scheduling would need large amounts of internal information like schedules, attendance, up-times of equipment.

ORGANIZING

Strategic organizing would need external and internal data to decide on re-structuring as will as forge strategic partnerships. Tactical organizing would need changing wage level data of both the organisation as well as that of competitors. Operational organising would need data relating to skills and training requirements of the operational staff.

#### CO-ORDINATION

Strategic coordination would call for industry wide data corresponding to technology availability. Tactical co-ordination would call for planet wide and supplier wise bottleneck data, which reflects the deficiencies both inside the organisation and outside. Operational co-ordination would require itemized break up of plant and machinery performance, failures etc.

#### DIRECTING

Strategic directing functions like introduction of office automation would call for detailed cost benefit analysis of new techniques. Tactical directing like innovating marketing strategy would call for detailed market and production data. Operational directing function would need data pertaining to the individual managers detailed skills.

### System Implementation

As specified in the software and hardware specification section in the beginning, since this system is built on the client server architecture, a central web server computer is required to manage the user requester services made operational. Since the web server is used Apache with Tomcat the system requirements listed by Apache with Tomcat holds good in this case also. Apart from the standard requirements of 128MB of RAM,1.7GB of hard disk space and a fast processor, a modem is required to set it up for the internet

A client on the other hand requires only a low end computer, even with 32 to 64 MB RAM and 1GB hard disk. Here too, the requirement of a modem is indispensable.

### 

### Maintenance

Software maintenance is the process of modifying a software system or component after its delivery in order to correct faults, improve the performance and other attributes, or to adapt to the changed environment.

Maintenance covers a wide range of activities including correcting the code and design errors, updating the documentation and test data, and upgrading the user support. There is an aging process that calls for periodic maintenance of hardware and software. Maintenance is always necessary to keep the system into its standards.

**SECURITY MEASURES TAKEN**

The Security Measures is the most important task, which should be done with much care no need to say why this is so crucial. The security measures starts right from the selection of operating system in that case for server unix is best choice because it provides best security in its peer OS as we know this project is very versatile and there are variety of users who uses systems according to their needs so authentication is an major task which has to perform by the system administrator as I have already mentioned that there are different type of access permission like viewing the records modifying etc.

Another security measure I took in this project is **username and password** provide to selected staff so that no unidentified person can not access the system.

Then I switch the concept of **file sharing** which is an essential service in these types of companies. Particular type of file can be shared by the person holding the same position.

**COST ESTIMATION OF THE PROJECT**

In the traditional engineering disciplines where one has to budget for material the “Item material” used in software is mainly the engineer’s brainpower.

**Thus the cost of a software project is directly proportional to the number of engineers need for the project.**

The problem of predicating how many engineers and other resources are needed for a given software project is known as software cost estimation.

Forecasting how many engineers will be needed is a difficult problem that is intimately tied to the problem of how to estimate the productivity of software engineers.

There are two parts to the forecasting problem:

* Estimating the difficulty of the task
* Estimating how many tasks each engineer can solve.

Clearly, to estimate difficulty of the task, one must know what the task is – i.e. , what the requirements are. But, it is often difficult to specify the software requirements completely.

It was precisely such difficulties that motivated me to look at the evolutionary process as a alternative to the traditional water fall model.

Incomplete and imprecise requirement hinders accurate cost estimation. The clearer and more complete the requirements, the easier it is to determine the resources required. But even when the requirements are clearer, estimating the number of engineers needed is a formidable task with inherit difficulties.

The best approach is to develop the resource requirements incrementally, revising current estimates, as more information becomes available.

Cost Estimation

Number of Employees : One

Number of months for

Completion of the Project : 4 months

Salary for Employee : 5000 INR

Cost of Employee Work : 15000 INR

Cost of Development : cost of man work + misc. cost

: 15000 +

: 18000 INR

So the estimated cost of this Project is around 18,000 Indian Rupees.

**PERT CHART FOR THE PROJECT**

A PERT **(Program Evaluation and Review Technique)** chart is a network of boxes (or circles) and arrows. There are different variations of PERT charts. The boxes in PERT Chart can be decorated with starting and ending dates for activities; the arrows help in computing the earliest possible starting dates for the boxes at their heads. Some boxes can be designated as milestone.

The Chart below shows that the path through the project that consists of the “design”, “build code generator”, and “integration and testing” activities is the critical path for the project. Any delay in any activity in this path will cause a delay in the entire project.

**The Diagram of PERT Chart of the Project is given on the following page.**

27 Feb, 17

Build Menu Frame

20 Jan, 17 10 Feb, 17 4 Mar, 17 18 Apr, 17 28 Mar, 17 5 May, 17 12 Apr, 17

Build Dialog Boxes

Write Manual

Finish

Build All Classes

Integration & Test

Design

Start

**GANTT CHART**

Gantt charts are a project control technique that can be used for several purposes, including scheduling, budgeting, and resource planning. A Gantt chart is a bar chart, with each bar representing an activity. The bars are dItemn against a time Line. The length of each bar is proportional to the length of time planned for the activity.

A Gantt chart helps in scheduling the activities of a project, but it does not help in identifying them. Gantt charts take different forms depending on their intended use. They are best for resource scheduling.

**The Diagram of Gantt Chart for this project is given on the following page.**

Start

**Jan 20,15 Feb 27,15 Mar 4,15 Apr 12,16 Apr 18,16 May 5,16**

Build All Classes

Build Dialog Boxes

Finish

Write Manual

Integration & Test

Build Menu Frame

Design

**Conclusion**

The software ‘MUSIC STORE MANAGEMENT’ reduces the considerable drawbacks like burden of human labor, portable defect and errors. This software saves time and provides 24 hour accessibility even from a remote place. Programs are menu driven which help even a newcomer to use the system with little training. Testing has been done with actual data and system is much better than the existing one. GUI makes the interface very much user friendly.

The system is highly user friendly and is well efficient to ease interactions with the users of the system. Reports generate with live data are proved to be informative and also helpful in making important decisions. The system is tested and implemented with high degree of accuracy.

The system is done with an insight into the necessary modification that may require in the future. Hence the system can be maintained successfully, without much rework.

**Scope for Future Enhancement**

As changes are always necessary in future it applies to software development also but these changes should be appreciable in nature. These appreciable changes will make the software to fight for its survival in the competitive market. Hence it is necessary to think about the future enhancements at present.

The system ‘MUSIC STORE MANAGEMENT’ will fulfill the entire requirement of the clients. The system is developed according to the present requirements of the company. The system is developed as easy as possible for the sake of end users.

One drawback of my system is that the client cannot view, search and purchase music according to a particular language option .By the next time I would like to add this facility.

By the next time I would like to add two more modules: Purchase Module and Accounting Module. Purchase Module deals with purchasing activities of music related items. Accounting Module deals with all accounting activities such as billing, ledger preparation, balance sheet preparation, profit and loss account preparation etc.

In the present system transaction is through a particular bank or through money orders. In future I would like to make it through credit cards. Credit card validation techniques are needed for that.

The developed software for the organization is flexible and it can be made to run on all kinds of platforms. The system is error free and highly portable. It can be implemented in any servers in the Internet providing an easy access to the clients. It also has more options of the future developments.

**Bibliography**

**Serial No. Title**

**1. Servlet Programming 2. Java Server Programming**

**3. Java Server Pages**

**4. Oracle8i**

**Online Resources**

* + - **http://www.google.com**
    - **http://oracle.com**
    - **http://Redbooks.IBM.com**
    - **http://javaguru.com**
    - **http://www.sun.com**