

## CHAPTER 1

### INTRODUCTION

The focus of our project was to add music system and to review the whole library entry system and improve upon it. We added the music system that are created a method to enter it into the existing database.

We are adding the music with respect to their film names order. We put the music system into boxes and learned ways to make new entries into the database. In order to learn more about how to better organize the library in the future. In addition, we found several websites that are also helpful for referencing our music library and to gather information for future recommendations. The project team decided to implement the core functionality first and later to attach additional functions. The Core functionality is:

- Add User information
- Add, delete, and update information about the album and songs.

#### 1.1 Related work

The related work is divided into four parts:

- Adding of music and images
- Display music and update music
- List of music galleries
- User login information

In the first case we add the movie names along with their music's and images into server database. This is done by selecting the albums, movie names and the Images. After that enter the album, title, images. When you are all done, click, upload.

Once the music's are uploaded we can display the music's and also, we can update here. The uploaded music's are listed in the music gallery with all entered information. The last part is user login information. Here all the names of the users are listed according to login.

#### 1.2 Proposed work

The proposed system is a web based system. The user can log in to the system with username shared to the music library. Then he/she can listen the favorite music

from favorite film. The administrator can login and manage the music library system. The proposed system uses GUI framework. This system is highly user friendly so that the new comer can use this webpage efficiently.

### **1.3 Objectives of the work**

- This project is designed as user friendly so that user can easily access their requirements about the music library.
- Admin will add the music and corresponding movies and related details for the album.
- Admin can login and view the user login information entered by the users and also, thereby improving the quality of music considering the requirements of user.

### **1.4 Key Features**

- The music library will enable the users to perform activities such as playing the music and view the latest added music.
- The music library will enable the users to select the music according to their favorite movies.
- The music library will have an admin page, which will enable admin to maintain user details and manage music items in the system.
- The application will allow the user to find the music's by their names, movie name and languages.

### **1.5 Organization of the Project**

This project report has been broadly divided into six chapters:

- **CHAPTER 1** gives propose of system, project motivation, problem statement, methodology used.
- **CHAPTER 2** discusses about literature survey.
- **CHAPTER 3** system requirement specification of the project.
- **CHAPTER 4** gives the system design of the project.
- **CHAPTER 5** gives the implementation details of the project.
- **CHAPTER 6** gives the results of the project.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 Music Information Retrieval System using Lyrics and Melody Information**

Spectrogram factorization methods such as Nonnegative Matrix Factorization (NMF) are frequently used as a way to separate individual sound sources from complex sound mixtures. More recently, they have also been used as a first stage for the automatic transcription of polyphonic music. The problem of sound source separation is different (but related) to the problem of automatic music transcription. The output of the first is the separated audio signals corresponding to each sound source, whereas the output of the second is a symbolic representation/music score that encodes the discrete pitches/notes that are played and when they are played. Many variations of factorization methods have been proposed. Two important design choices are the way spectra are represented and what distance measures are used to compare them in the optimization used for factorization. A common assumption has been that a variant that yields better signal separation will result in better automatic transcription. In this work, we investigate experimentally this question and show that this relationship is not necessarily true. Index Terms—Non-negative matrix factorization, Music Transcription, Beta-Divergence, Sound Source Separation.

#### **2.2 Musical Onset Detection on Carnatic Percussion Instruments**

In this work, we explore the task of musical onset detection in Carnatic music by choosing five major percussion instruments: the mridangam, ghatam, kanjira, morsing and thavil. We explore the musical characteristics of the strokes for each of the above instruments, motivating the challenge in designing an onset detection algorithm. We propose a non-model based algorithm using the minimum-phase group delay for this task. The music signal is treated as an Amplitude-Frequency modulated (AM-FM ) waveform, and its envelope is extracted using the Hilbert transform. Minimum phase group delay processing is then applied to accurately determine the onset locations. The algorithm is tested on a large dataset with both controlled and concert recordings (tani avarthanams). The performance is observed to be the

comparable with that of the state-of-the-art technique employing machine learning algorithms.

### **2.3 Factors in factorization: Does better audio source separation imply better polyphonic music transcription**

Multimedia content can be described in versatile ways as its essence is not limited to one side. For music data these multiple fields could be a song's audio features as well as its lyrics. But most recent research revolves around melody information for retrieval. Therefore, we proposed an MIR system that utilizes the user's acoustic signal from a singing voice and retrieves the music information using both lyrics and melody information. The lyrics recognition module uses a keyword spotting system based on textcontent of the lyrics by an HMM comparison engine. The melody recognition module extracts pitch and MFCC features from the user singing input and then retrieves music by a GMM comparison engine. Consequently, the proposed MIR system consists of fusing the lyrics and melody recognition module in which the melody recognition especially operates to restrict recognition candidates. Experiments show that the proposed MIR system has recognition rate of 72.72% to 83.64% when the numbers of restricted recognition candidates are from 10 to 50.

### **2.4 Review Mining for Music Digital Libraries**

We continue our work on the automatic mining of user-created music reviews towards the goal of connecting user opinions to music objects in Music Digital Libraries (MDL). We demonstrate an experimental system which automatically discovered the key descriptive patterns that differentiated positive from negative reviews which helps us to better understand our successful Phase I results. Comparison to an earlier study indicates an important consistency across projects that warrants further investigation.

## CHAPTER 3

# SYSTEM REQUIREMENT SPECIFICATION

### 3.1 System Analysis

System analysis will be performed to determine if it is feasible to design information based on policies and plans of the organization and on user requirements and to eliminate the weaknesses of the present system

- To enhance user / system interface.
- To improve information quality and usability.
- To upgrade systems reliability, availability, flexibility and growth potential

### 3.2 Functional Requirement

- It will list out the minimum requirements required for the project to have.
- User has to login with his username.
- One account cannot be associated with multiple users.
- System should only allow users to move to library only when mandatory fields such as movie names, language.

### 3.3 Non-Functional Requirement

The following are the system requirements for music library systems

#### HARDWARE SPECIFICATION

Processor	: Intel P4 CPU 1.60 GHz
Memory	: 128MB
Hard Disk	: 20GB

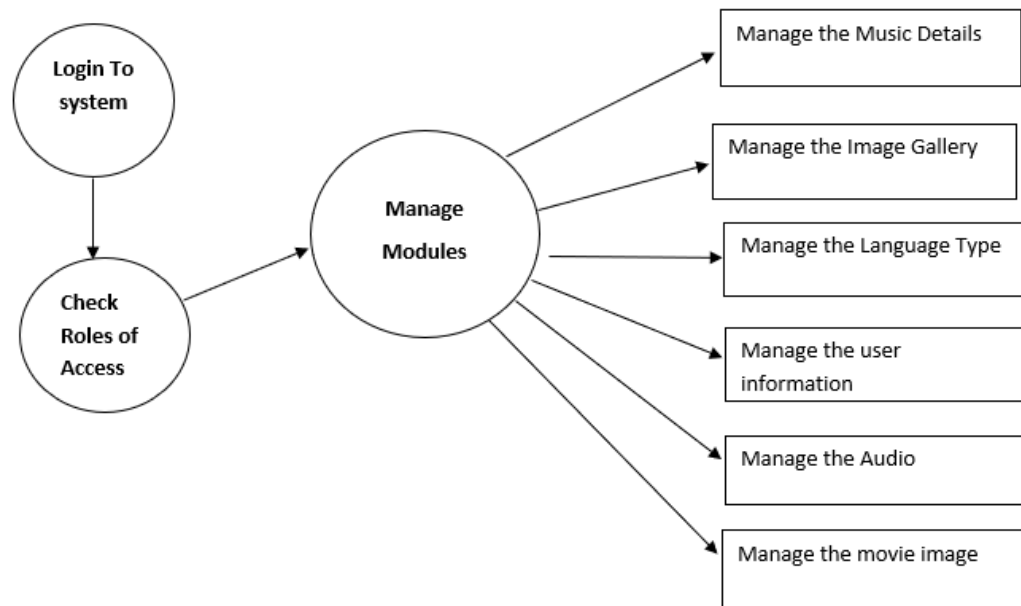
#### SOFTWARE REQUIREMENTS

Operating system	: Windows XP or later
Front end language	: PHP, HTML, CSS, JAVA Script
Back end/Database	: MySQL
Development tools	: XAMPP, Notepad++
Browser	: Any browser

## CHAPTER 4

### SYSTEM DESIGN

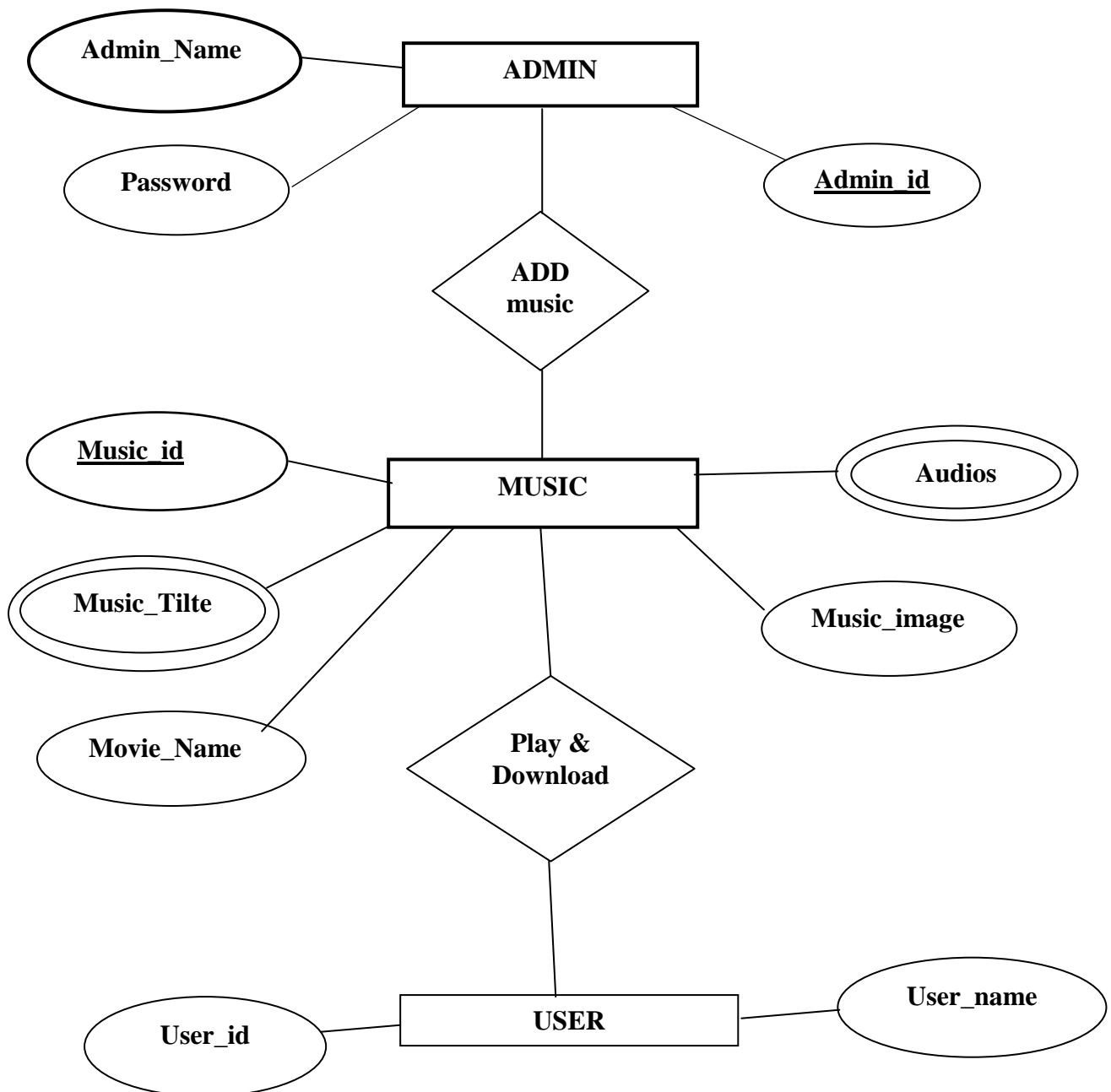
#### 4.1 Work model



**Fig 4.1 work model**

## 4.2 Entity Relationship Model

An entity relation model (ER model) describes inter-related things of interest in a specific domain of knowledge. It is composed entity types and specifies relationship that can exists between instance of those entity types. It is usually drawn in a graphical form as boxes(entities) that are connected by lines (relationship) which express the association and dependencies between entities.



**Fig 4.2** ER diagram

## 4.3 Database Design

### Database name with attributes

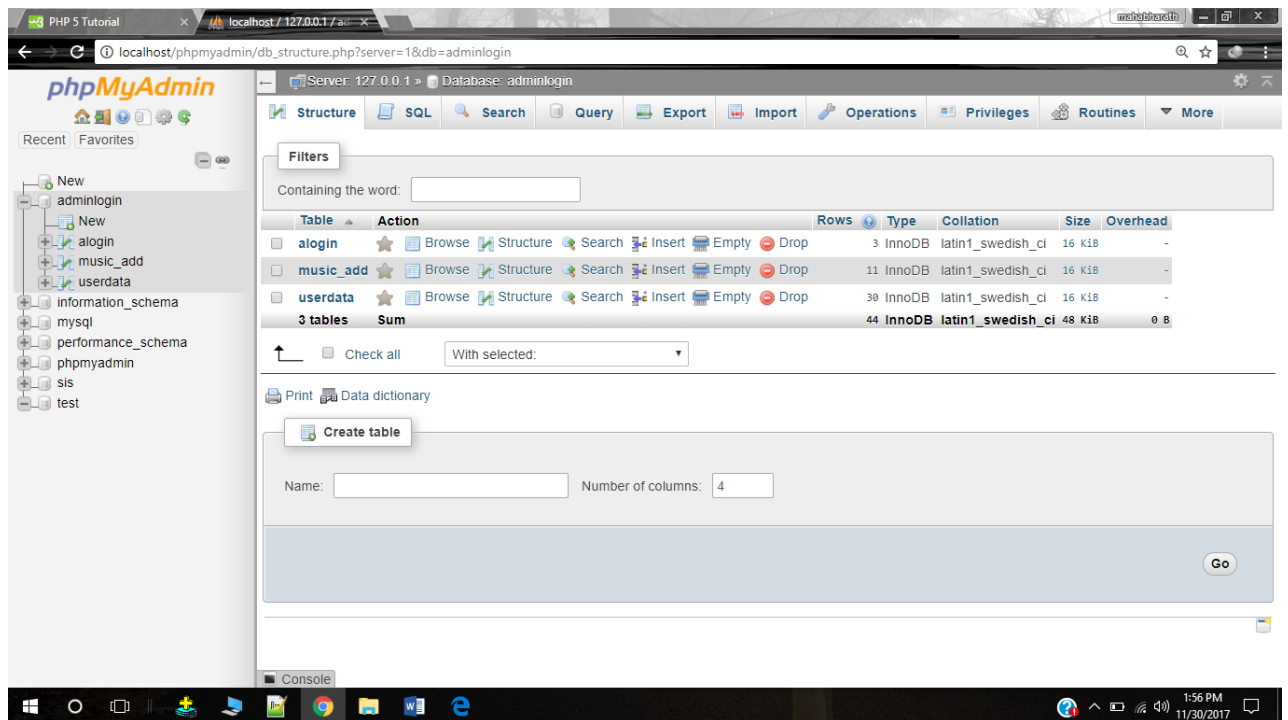


Fig 4.3.1 database attributes

### Admin login

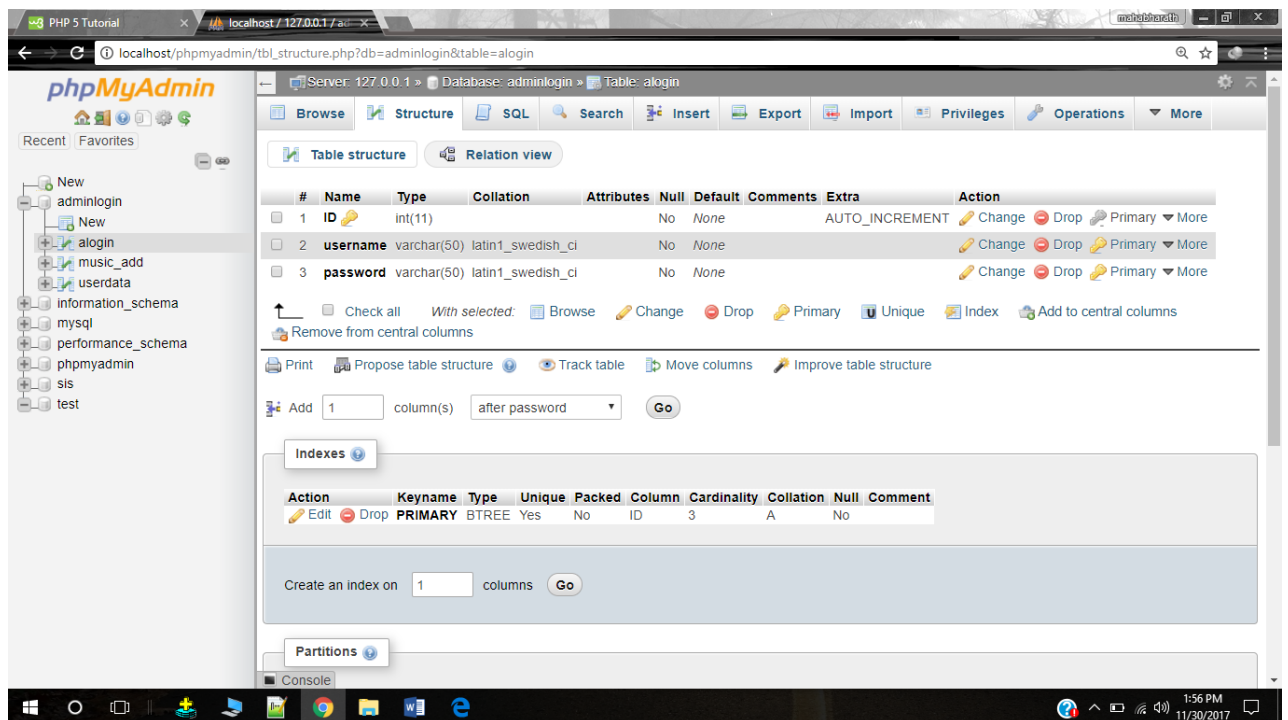


Fig 4.3.2 admin login attributes screenshot



## Add music :

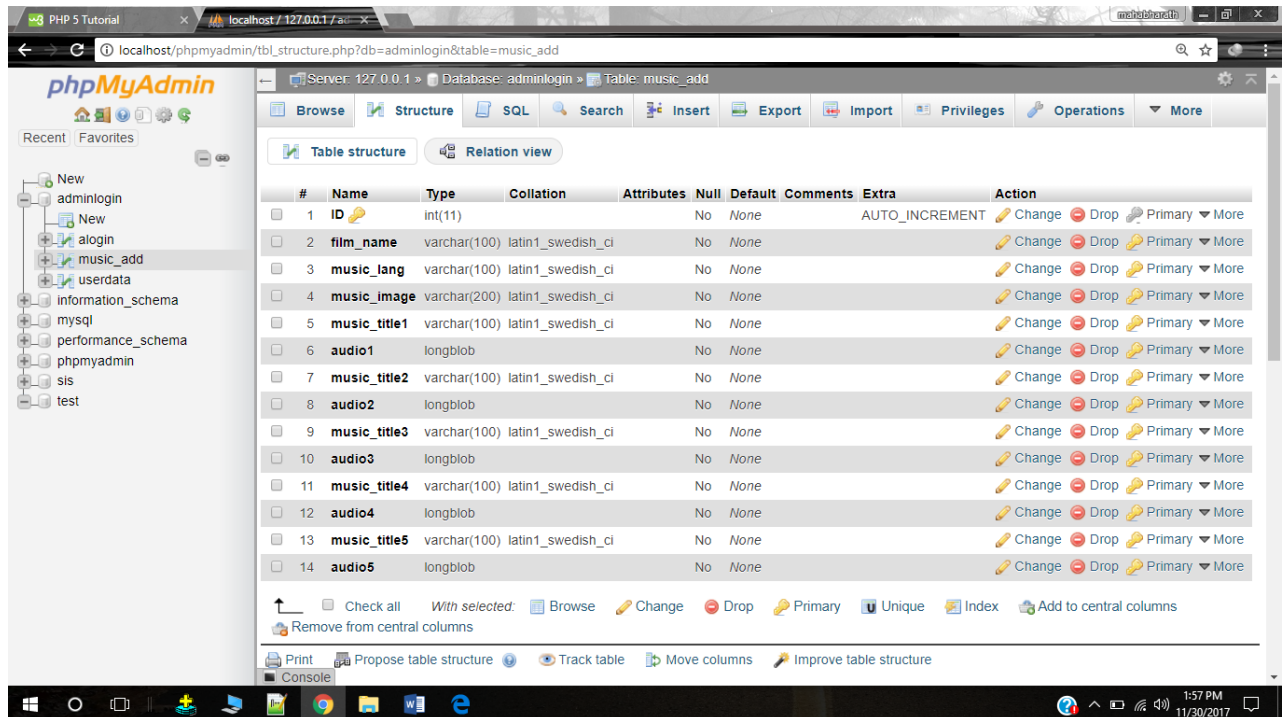


Fig 4.3.3 add music database screenshot

## User login :

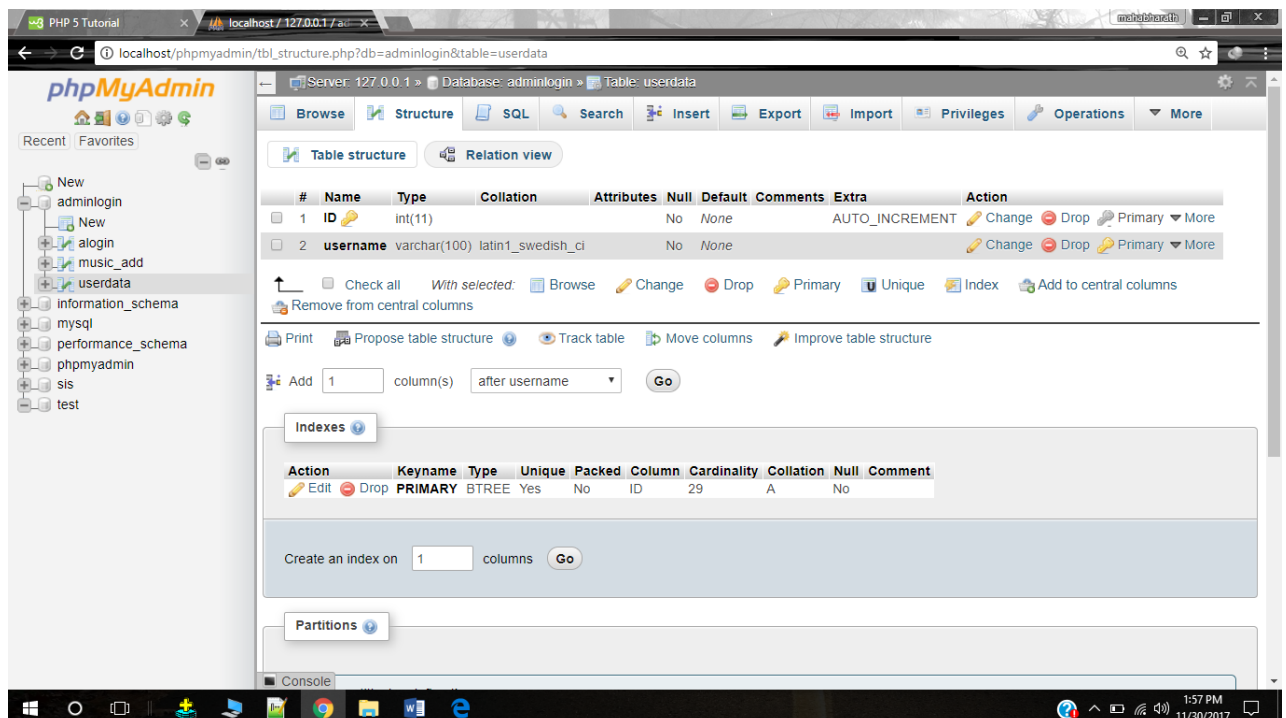


Fig 4.3.4 user login database screenshot

## CHAPTER 5

# IMPLEMENTATION

This section deals with the implementation details of the system designed in the previous section. Implementation is the process of converting design to code. The entities identified from design are to be implemented considering the association between them and how they communicate with each other.

### 5.1 Entities

Here we discuss about the modules implemented in the system. Important aspects of the system are discussed. The implementation is divided into module wise so that it is easy to implement. Implementation is mainly done via three modules:

- ALOGIN
- ADD MUSIC
- USER DATA

### 5.2 Entities description

Each entity action is described in the following section

#### 5.2.1 ALOGIN Entity

This entity has the main responsibility of the Music library. This admin has the following pages

- The Add music page
- The Modify music page
- The View user login page
- The view gallery page

#### **The Add Music Page:**

This page enables admin to add music to libraries.

#### **The Modify Music page:**

This page enables the admin to manage and modify the music in libraries.

#### **The View user login page:**

This page enables the admin to view the number of user's login in the music libraries.

#### **The view gallery page:**

This page enables the admin to view the number of music added in the music libraries.

### 5.2.2 ADD MUSIC

This entity has the main responsibility to add Music in music library. This allows the admin to add movie name, movie language and music into the library.

### 5.2.3 USER DATA

This entity has the main responsibility to view the user login information in the admin entity. User can view all the music that are currently available in the music library.

## 5.3 Pseudo Coding

Let us see sample coding displayed here. As we cannot give the full coding here, we will see about

### Home.php

```
<?php
session_start();
if($_SESSION["admin"]=="")
{ ?>
<script type="text/javascript">
    window.location="mylogin.php";
</script>
<?php
}
include "header.php";
include "link_db.php";
?>
<link href="pagination/css/pagination.css" rel="stylesheet" type="text/css">
<link href="pagination/css/A_green.css" rel="stylesheet" type="text/css">
<section id="slider">
<div class="container">
<div class="row">
    <div class="col-sm-12">
        <div id="slider-carousel" class="carousel slide" data-ride="carousel">
            <ol class="carousel-indicators">
                <li data-target="#slider-carousel" data-slide-to="0" class="active"></li>
                <li data-target="#slider-carousel" data-slide-to="1"></li>
                <li data-target="#slider-carousel" data-slide-to="2"></li>
            </ol>
<div class="carousel-inner">
            <div class="item active">
                <div class="col-sm-6">
                    <h1><span>My</span>-MUSIC</h1>
                    <h3><b>"Music gives a soul to the universe, wings to
the mind, flight to the imagination and life to
everything"</b></h3>
```

---

```

        </div>
        <div class="col-sm-6">
        
        </div>
        </div>
        <div class="item">
        <div class="col-sm-6">
        <h1><span>My</span>-MUSIC</h1>
        <h2>"You can't touch music but music can touch you"</h2>
        </div>
        <div class="col-sm-6">
        
        </div>
        </div>
        <div>
        <div>
        <div class="item">
        <div class="col-sm-6">
        <h1><span>My</span>-MUSIC</h1>
        <h2>"Where worlds fail MUSIC speaks" </h2>
        </div>
        </div>
        <div class="col-sm-6">
        
        </div>
        </div>
        </div>
        <a href="#slider-carousel" class="left control-carousel hidden-xs" data-slide="prev">
        <i class="fa fa-angle-left"></i>
        </a>
        <a href="#slider-carousel" class="right control-carousel hidden-xs" data-
        slide="next">
        <i class="fa fa-angle-right"></i>
        </a>
        </div>
        </div>
        </div>
        </div>
        </section><!--/slider-->
        <div class="col-sm-9 padding-right">
        <div class="features_items"><!--features_items-->
        <h2 class="title text-center">Top Movies</h2>
        </div>
        <?php
        include("pagination/function.php");

        $page=(int) (isset($_GET["page"]) ? 1 : $_GET["page"]);
        $limit = 9;

```

```

$startpoint = ($page * $limit)-$limit;
$statement = "music_add";
$res=mysqli_query($link,"select * from {$statement} LIMIT {$startpoint}, {$limit}");

while($row=mysqli_fetch_array($res))
{
    ?>
    <div class="col-sm-4">
    <div class="product-image-wrapper">
    <div class="single-products">
    <div class="productinfo text-center">
    " width="300"
height="168" alt="" />
    <h2><?php echo $row["film_name"]; ?></h2>
    <a href="show_music.php?ID=<?php echo $row["ID"]; ?>" class="btn btn-
default add-to-cart">Show Music</a>
    </div>
    </div>
    </div>
    </div>
    <?php
    }
    ?>
    </div>
    <ul class="pagination">

    <?php
        echo pagination($statement,$limit,$page);
    ?>
    </ul>
    </div>
    </div>
    </div>
    </div>
</section>
<?php

include "footer.php";
?>

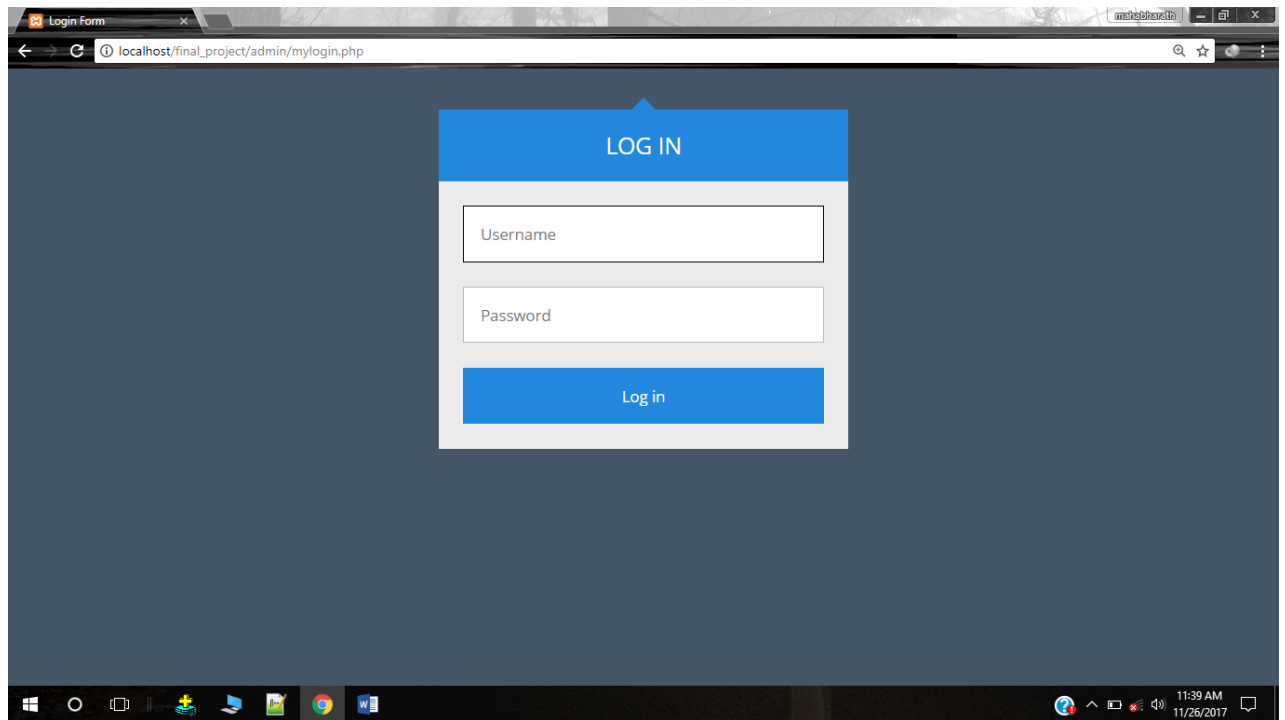
```

## CHAPTER 6

### RESULTS AND DISCUSSION

The results are verified by observing the input and output. Following are the screen shots of the Music Library System.

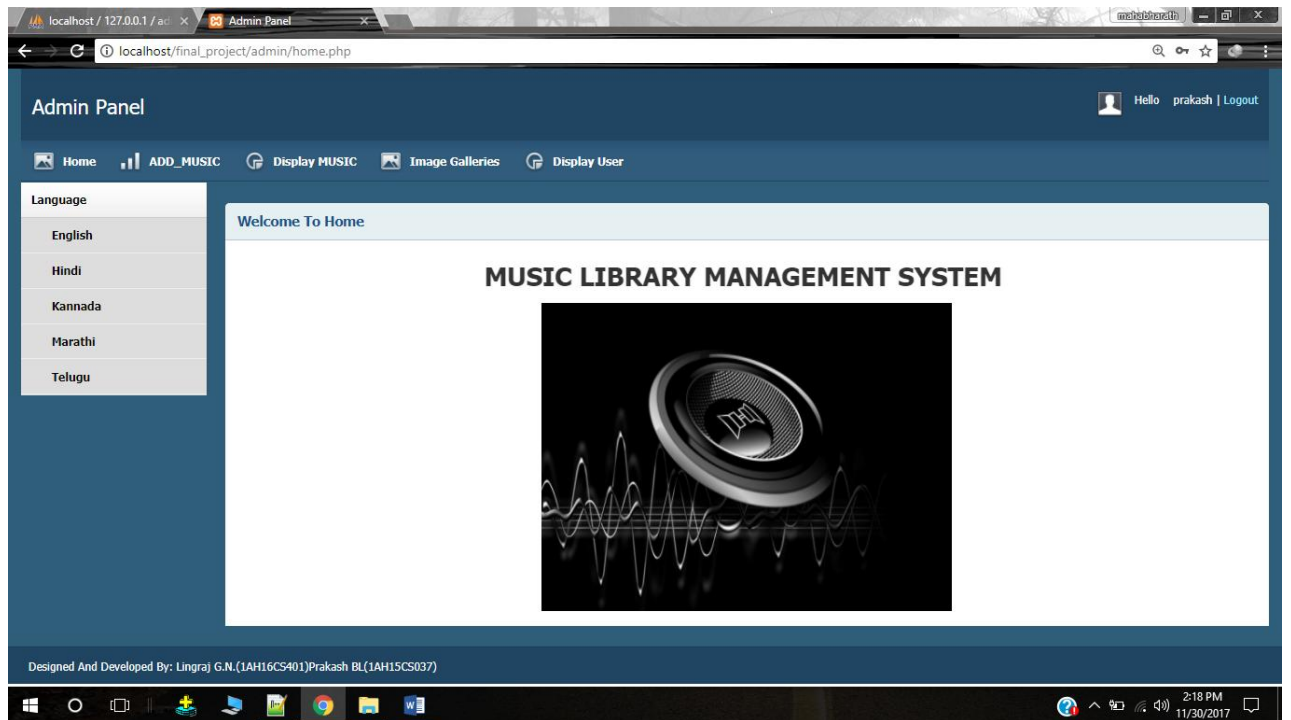
#### Admin login page



**Fig 6.1** Admin login page screenshot

A **login** generally requires the admin to enter two pieces of information, first a user name and then a password. If username and password is correct then only he can access the admin panel.

## Admin home page



**Fig 6.2** Admin home page screenshot

After the login, home page is displayed for admin.

## Add music page

Admin Panel

Notice: A session had already been started - ignoring session\_start() in C:\xampp\htdocs\final\_project\admin\header.php on line 2

Hello prakash | Logout

Home ADD\_MUSIC Display MUSIC Image Galleries Display User

Language

English

Hindi

Kannada

Marathi

Telugu

ADD\_MUSIC

Movie Name::

Music Language:: Select Language

Music Image:: Choose File No file chosen

Audio1:: Choose File No file chosen

Audio2:: Choose File No file chosen

Audio3:: Choose File No file chosen

Audio4:: Choose File No file chosen

Audio5:: Choose File No file chosen

upload

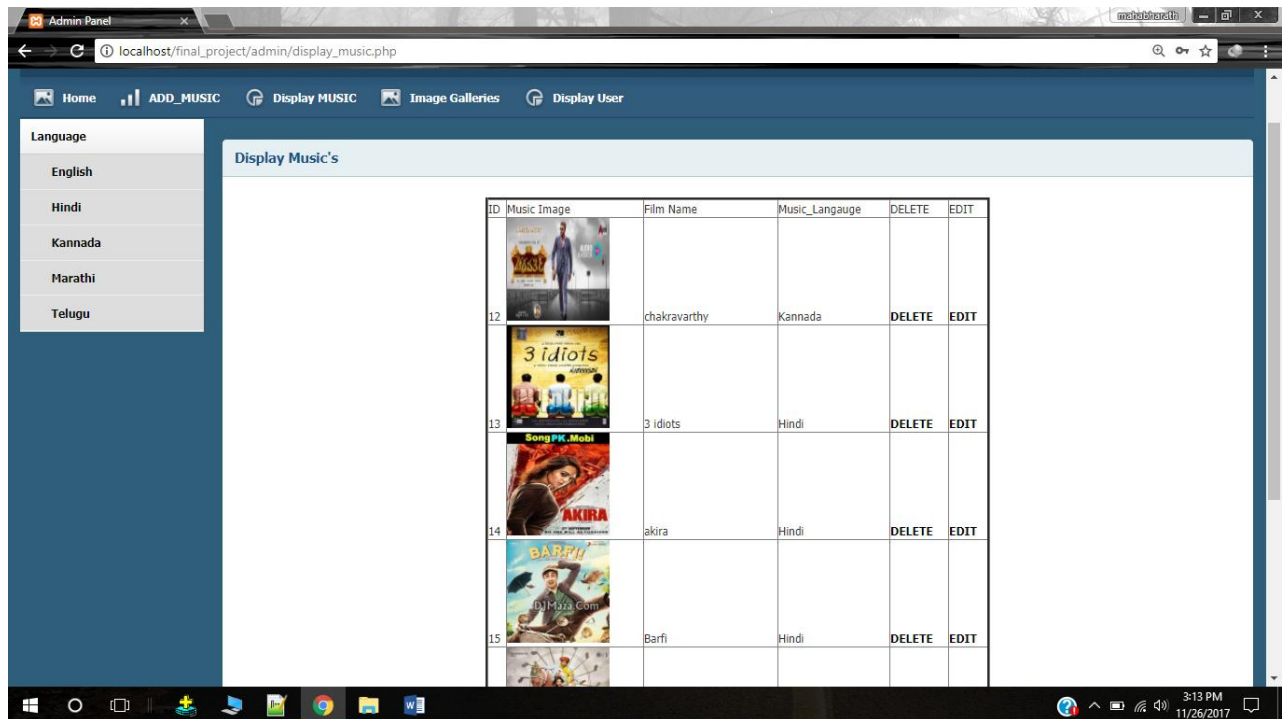
Designed And Developed By: Lingraj G.N.(1AH16CS401)Prakash BL(1AH15CS037)

**Fig 6.3** Add music screenshot

Here admin add the music to music library with movie name, music language and audios.



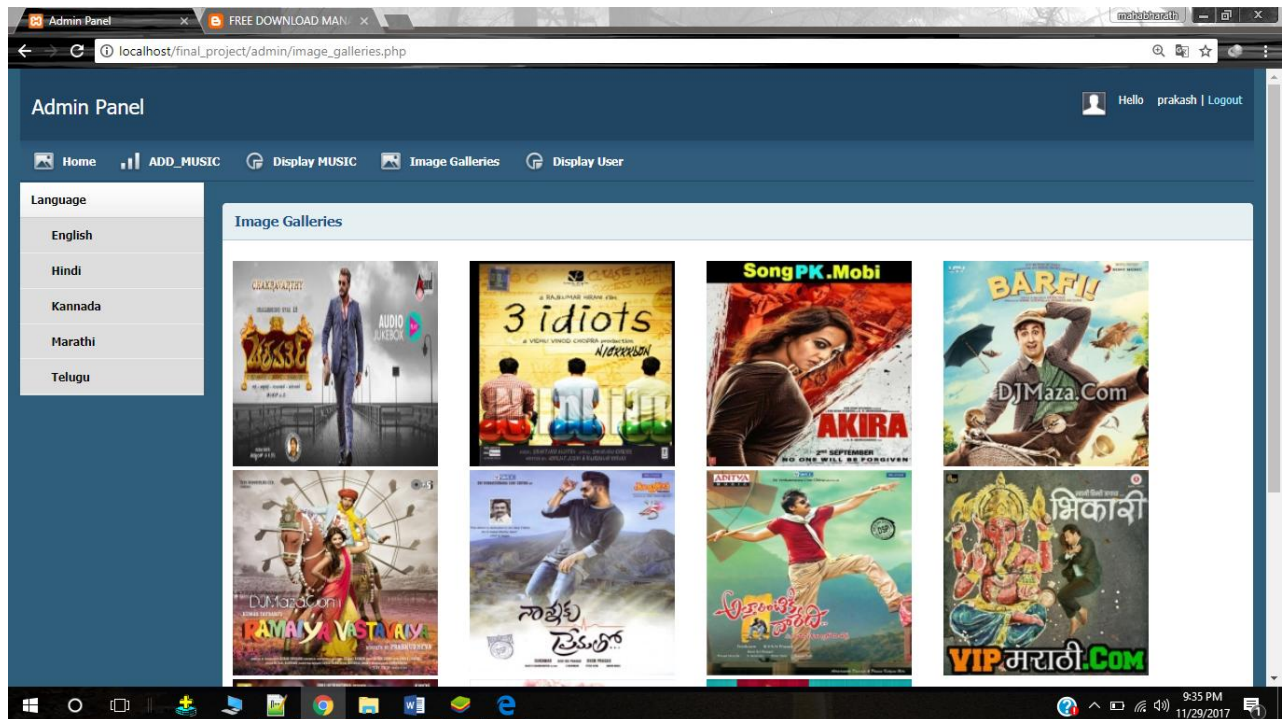
## Display music



**Fig 6.4** Display Music Screenshot

In this panel admin can view the musics with image, film name, language and he can delete and edit the musics.

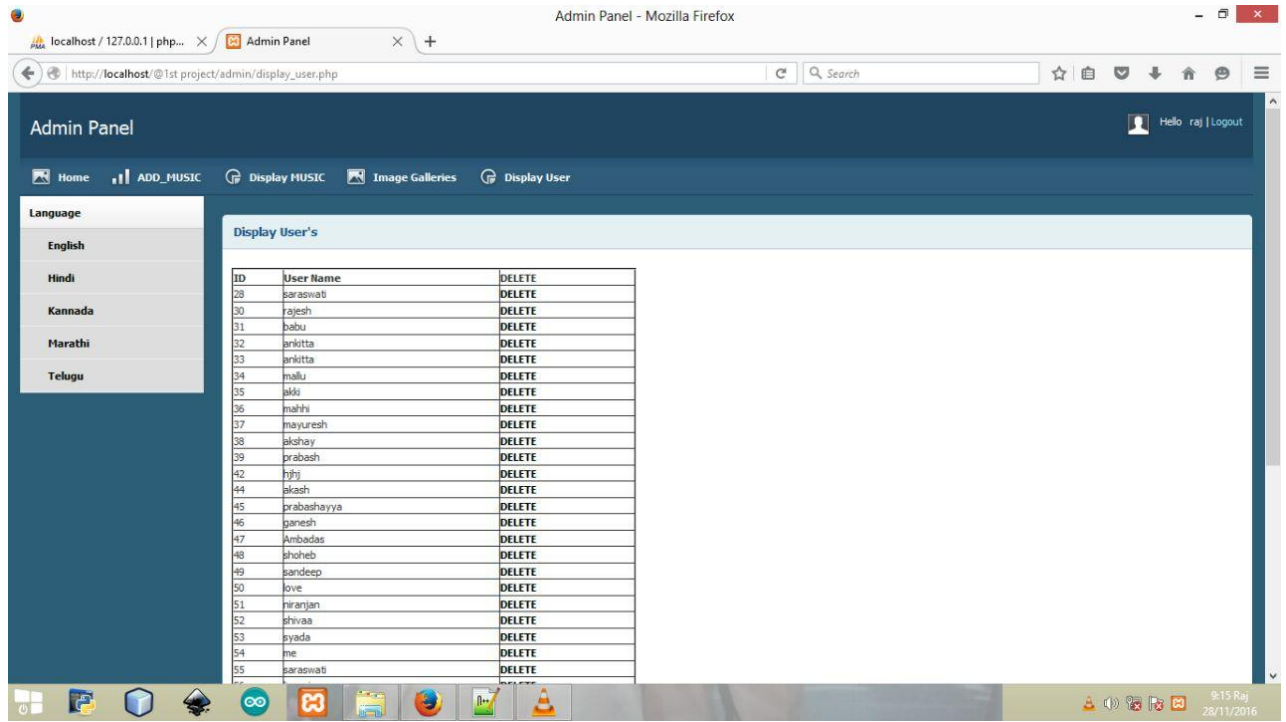
## Image gallery



**Fig 6.5** Image Galleries Screenshot

This page enables the admin to view the number of music added in the music libraries with images.

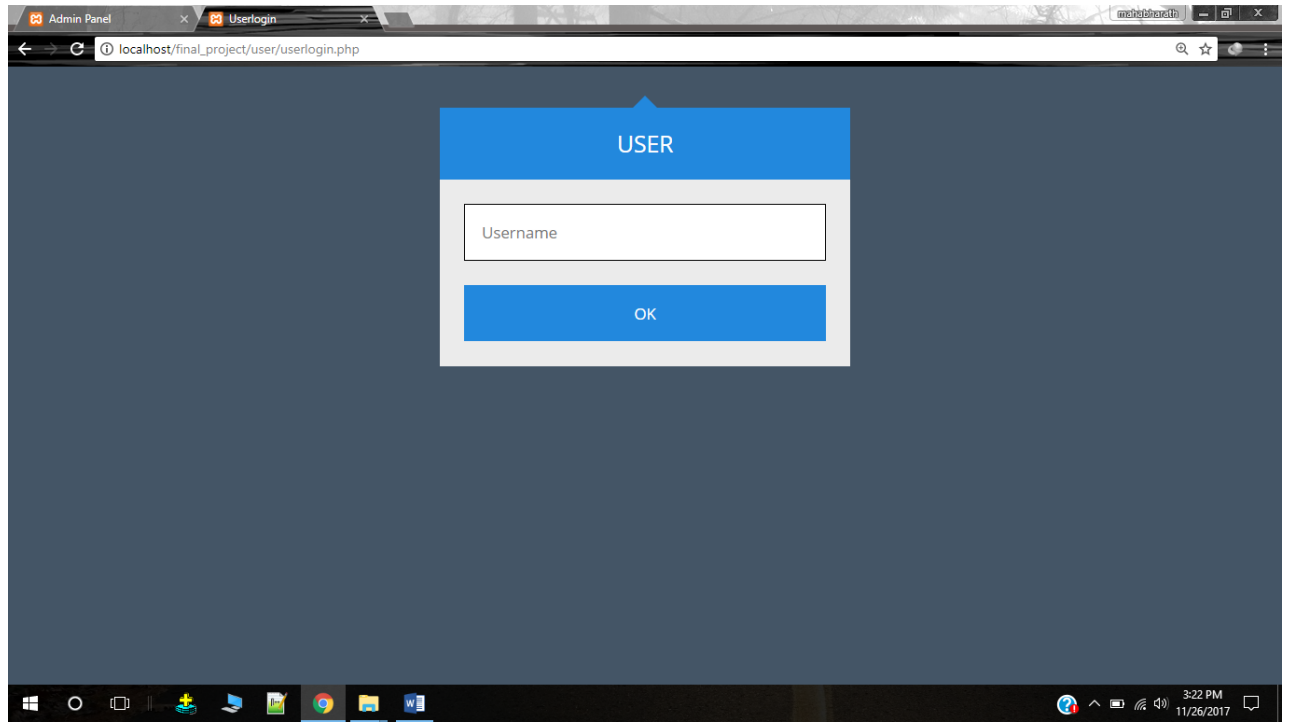
## Display user:



**Fig 6.6** Display User Screenshot

User names are displayed to an admin when the user logs on to the music library and the admin can manage the user information (i.e. he can delete the user name).

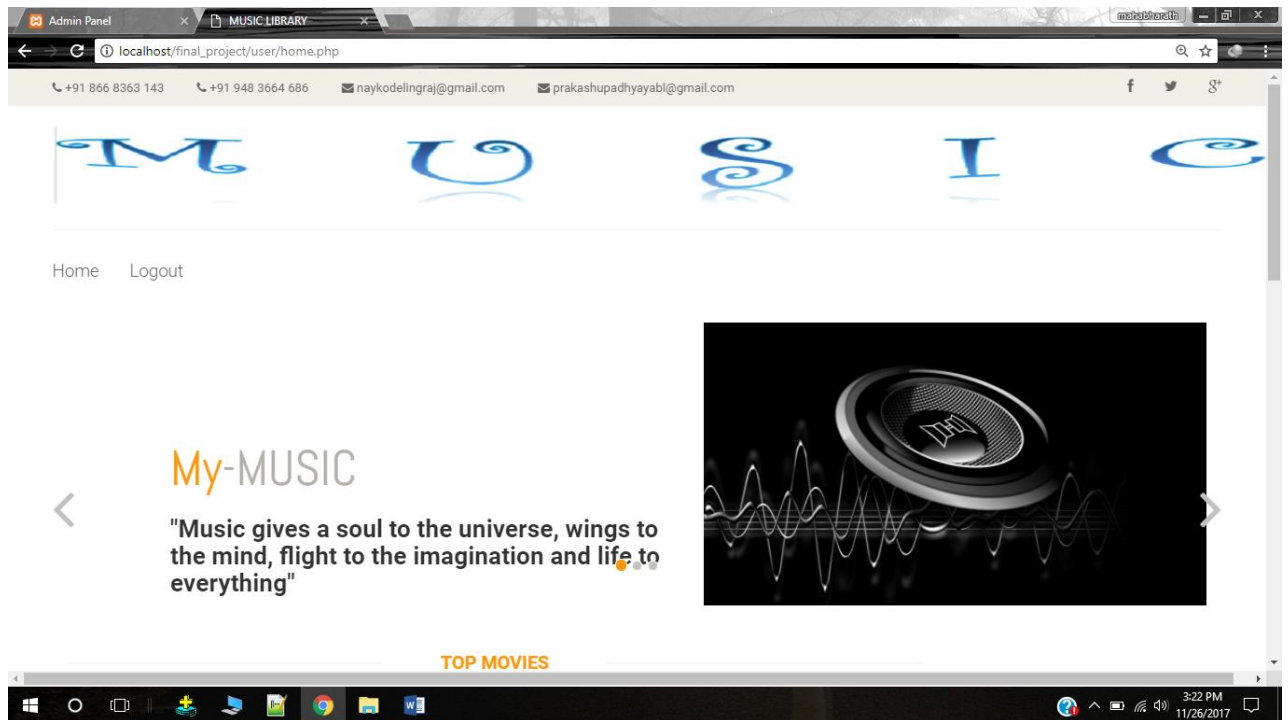
## User login page:



**Fig 6.7** User Login Page Screenshot

To logon to the music library, users will provide logon information in the login page. Whether he is user or administrator is determined based on the user name.

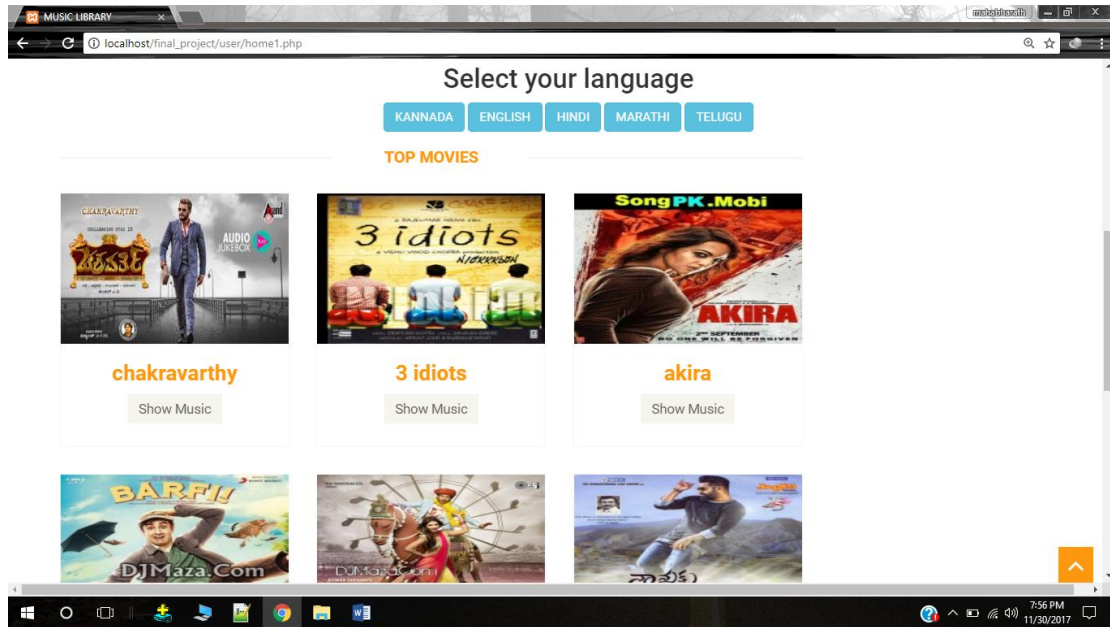
## User home page:



**Fig 6.8** User home page screenshot

The home page of the music library will be the first page to be displayed when a user visits the music library. The page will display a message and, in addition, it display various menus.

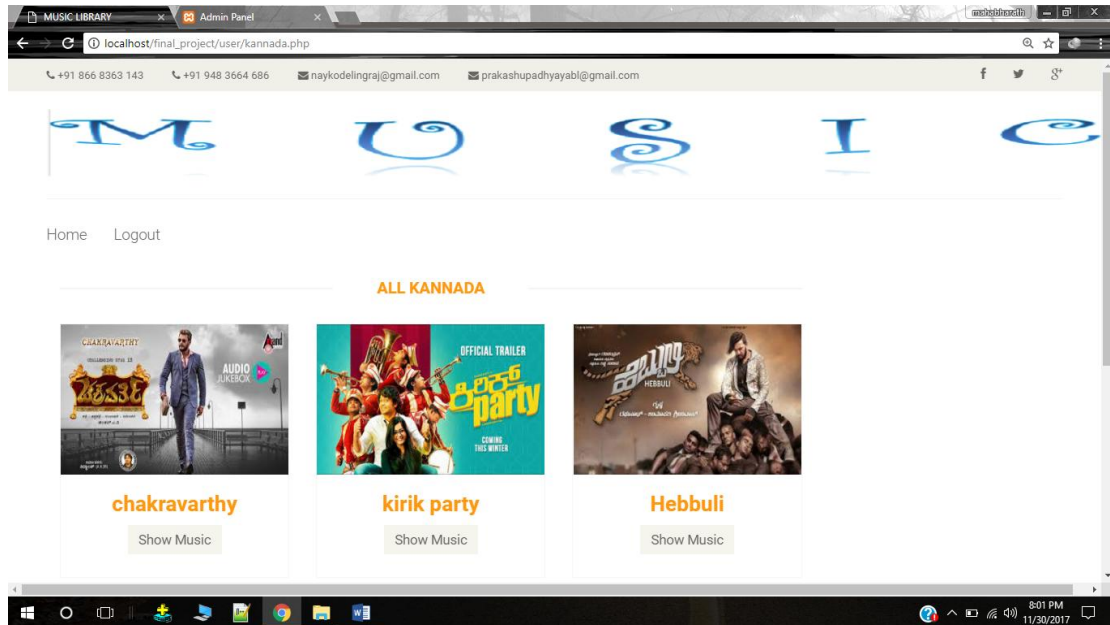
## Home page with albums



**Fig 6.9** User home page with albums screenshot

In this page musics are displayed according to the insertion and user can choose the musics according to languages.

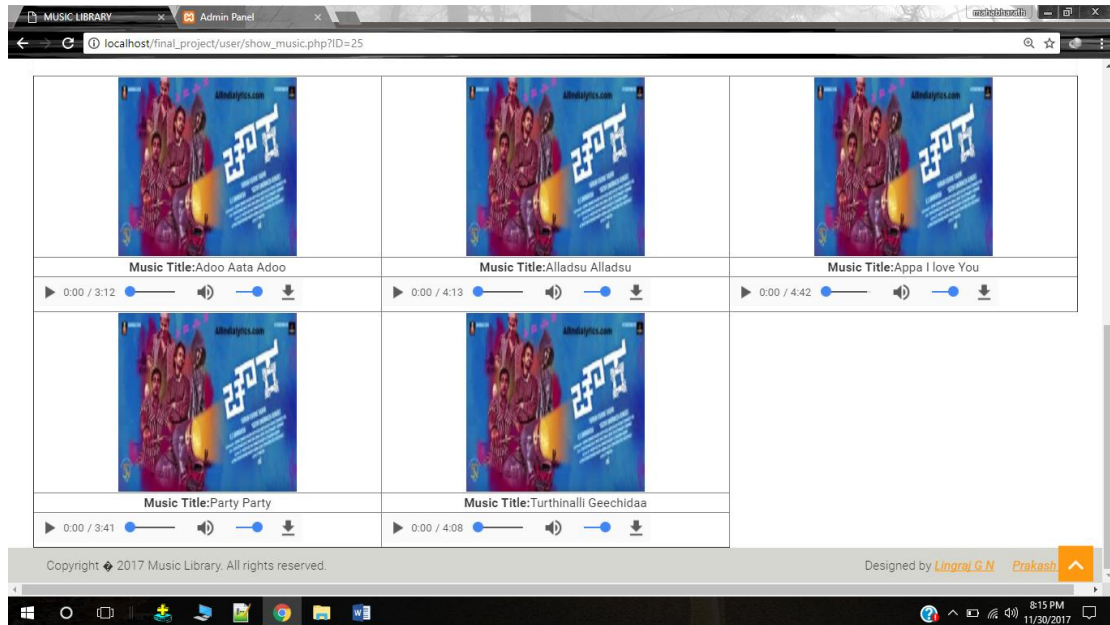
## List of movies from selected language



**Fig 6.10** list of movies from selected language

After selecting the user's favorite language, after clicking the language button number of movies of selected languages are displayed. In this page user can choose his favorite movie also.

## Music from selected movie



### 6.11 Music from selected movie screenshot

After enter the home page user can choose any one movie. After selecting the movie it displays the number of musics of that movie.



## **CONCLUSION**

The music library system reduces the considerable drawbacks like burden of human labor, portable defect and errors. This system saves time and provides 24 hours accessibility even from a remote place. Programs are easy which help even a new comer to use the system easily. GUI makes the interface very much user friendly.

The system is highly user friendly and as well efficient to interaction with the users of the system. The system is implemented with high degree of accuracy. The system is done with an insight into necessary modification that may require in the future. Hence the system can be maintained successfully, without much rework.

# REFERENCES

## JOURNALS

1. Music Information Retrieval System using Lyrics and Melody Information  
Tao Wang, Dong-Ju Kim and Kwang-Seok Hong School of Information and  
Communication Engineering, Sungkyunkwan University Suwon, Kyungki-do,  
440-746, Korea wtw12@skku.edu, radioguy@skku.edu.
2. Musical Onset Detection on Carnatic Percussion Instruments P A Manoj  
Kumar, Jilt Sebastian and Hema A Murthy Indian Institute of Technology  
Madras Chennai, India
3. Factors in factorization: Does better audio source separation imply better  
polyphonic music transcription Tiago Fernandes Tavares #1, George Tanti's  
#, Peter Driessen # # University of Victoria Department of Computer Science  
3800 Finnerty Road - Victoria, BC, Canada
4. Review Mining for Music Digital Libraries: Phase II J. Stephen Downie  
University of Illinois at Urbana-Champaign 501 E. D  
aniel St. Champaign, IL, 61820 jdownie@uiuc.edu

## BOOKS

6. PHP MySQL website programming problem-design-solution by Chris lea  
Mike Buzzard, Dileep Thomas, Jessy White-Clines.
7. Beginning Php5, Apache and MySQL Web Development by Elizabeth  
Naramore.

## WEBSITES

<https://www.slideshare.net>  
<https://www.w3schools.com/php>

