



**R. C. Technical
Institute
Sola, Ahmedabad - 60**

CERTIFICATE

This is to certify that this work of **PROJECT** Subject & **3360707** Subject Code of **6th Sem** with title: **MuCiX(website)** represents the work of the following students for the fulfillment of the Certificate of Diploma in Computer Engineering at R. C. Technical Institute Sola, Ahmedabad - 60, Gujarat, during the academic year **2021 to 2022** and the work is completed and found satisfactory.

Sr. No.	Enroll No.	Name
1	196400307098	Kalp Patel
2	196400307100	Khush Patel
3	196400307105	Manan Patel
4	196400307112	Samarth Patel

Submitted to: **HEMLATA V KATPARA**

Lecturer, computer Engineering Dept.

RCTI

HEMLATA V KATPARA

Head Of Computer Dept

ACKNOWLEDGMENT

We would like to express our special thanks of gratitude to our faculty and guide **Hemlata V. Katpara**, head of department **Hemlata V. Katpara**, who gave us the golden opportunity to do this wonderful project on the topic (**MUCIX Website**), which also helped us in doing a lot of Research and came to know about so many new things we are thankful to them. Secondly, we would also like to acknowledge our friends of R.C TECHNICAL INSTITUTE for their all kind of support and their suggestions. We are making this project not only for marks but also to learn something new and to increase our knowledge regarding new technologies.

THANKS AGAIN TO ALL WHO HELPED US

Regards

196400307098(Kalp Patel)

196400307100(Khush Patel)

196400307105(Manan Patel)

196400307112(Samarth Patel)

DETAILS OF CHAPTERS

1.0 <u>INTRODUCTION</u>
1.1 Project Summary & Features
1.1.1 Project Summary
1.1.2 Features
1.2 Purpose & Scope
1.2.1 Purpose
1.2.2 Scope
1.3 Objectives and Functionality
1.3.1 Objectives
1.3.2 Functionality
1.4 Technologies
2.0 <u>PROJECT MANAGEMENT</u>
2.1 Project Planning and Approach
2.1.1 Project planning and Approach
2.2 Project Scheduling
2.3 Roles and Responsibility
2.4 Project Development Approach
2.4.1 Iterative Model
3.0 <u>SYSTEM REQUIREMENT STUDY</u>
3.1 Proposed System
3.1.1 Modules & Features In the New System

3.1.2 User Characteristics
3.1.3 Hardware & Software Requirements
3.1.4 Assumptions and Dependencies
<u>4.0 SYSTEM ANALYSIS</u>
4.1 Technically Feasibility Study
4.2 Feasibility Study
4.2.1 Economic Feasibility Study
4.2.2 Technical Feasibility Study
4.2.3 Operational Feasibility Study
4.3 Hardware Study
4.4 Identify Customer's needs.
4.5 Perform Economical and Technical Analysis.
4.6 Use Case Diagram
<u>5.0 SYSTEM DESIGN</u>
5.1 Database Design/Data Structure Design
5.1.1 Data Dictionary (Tables and Relationship)
5.1.2 ER Diagram
5.1.3 Data Flow Diagram (Context Diagram, 0-Level and 1-level)
5.2 Input/output and Interface Design
5.2.1 Flow Chart
<u>6.0 IMPLEMENTATION</u>
6.1 Implementation Environment
6.2 Screenshots of Forms, Reports and Interface

6.3 Sample Coding
7.0 <u>TESTING</u>
7.1 Testing Plan & Strategy
7.2 Testing Methods
7.3 Test Cases
8.0 <u>SECURITY MEASUREMENTS AND FUTURE ENHANCEMENT</u>
9.0 <u>CONCLUSION / SUMMARY</u>
10.0 <u>REFERENCES / BIBLIOGRAPHY</u>

CH:1

INTRODUCTION

1.1 Project Summary & Features

1.1.1 Project Summary

1.1.2 Features

1.2 Purpose & Scope

1.2.1 Purpose

1.2.2 Scope

1.3 Objectives and Functionality

1.3.1 Objectives

1.3.2 Functionality

1.4 Technologies

1.1 PROJECT SUMMARY AND FUNCTION

1.1.1 PROJECT SUMMARY

- This is the music player website with the Interactive UI for the easy access for user. It can be used easily for music and it's convenient and quick, uses simple UI, in this music website we can play songs for free.
- The key features are as follows,
- This website integrates the Intent APIs of various applications and mapping the data from various databases which is to be segregated and they all are integrated onto our website.
- It can be used easily for music and it's convenient and quick, uses simple UI, in this music website we can play songs for free.
- We provide the High-Quality audio to user for free of cost.
- It provides functionality like creating a playlist and also has search functionality which will the search the song and give the desired result.
- Some things to enjoy in classical music Moods and feelings Loudness and softness Different speeds Instrument sounds Melodies Rhythms Changes and transformations beautiful performing Memories that get triggered Recognition of something heard earlier Visual images that come to mind.
- Streaming music, or more accurately streaming audio, is a way of delivering sound — including music — without requiring you to download files from the internet. Music services like Spotify, Pandora, and Apple Music use this method to provide songs that can be enjoyed on all types of devices.
- Instead of downloading multiple apps and wasting all storage space, this proposal reduces the use of a lot of applications and will be a complete music solution.
- Protecting the privacy and security of your information is our priority for everyone at “Mucix”.

1.1.2 FEATURES

- Features Of Online Music Portal System (MUCIX) Are as Follows:
- ✓ Provides the searching facilities based on various factors. Such as Music, Album, Customer, Album Type.
- ✓ The transactions are executed in off-line mode, hence on-line data for Music, Performer capture and modification is not possible.
- ✓ It tracks all the information of Performer, Track, Customer act Manage the information of Performer.
- ✓ It Shows the information and description of the Music, Album.

1.2 SCOPE

- Our project aims at business process automation i.e., we have tried to computerized various process of Online Music Portal
- It satisfies the user requirement
 - ✓ Be easy to understand by the user and the operator.
 - ✓ To utilize resource in an efficient manner by increasing their productivity through automation.
 - ✓ Be easy to operate.
 - ✓ Have a good user interface.

1.3 OBJECTIVES AND FUCTIONALITY

1.3.1 OBJECTIVES

- The main objective of the Online Music Portal is to manage: -
 - ✓ the details of Music,
 - ✓ Performer,
 - ✓ Album,
 - ✓ Customer,
 - ✓ Album Type.
- It manages all the information about Music, Track, Album Type.

1.3.2 FUNCTIONALITY

- Functionalities provided by Online Music Portal are as follows:
 - ✓ Provides the searching facilities based on various factors. Such as Music, Performer, Track, Album Type.
 - ✓ Online Music Portal also manage the Customer details online for Track details, Album Type details, Music.
 - ✓ It tracks all the information of Album, Customer, Track etc.
 - ✓ Manage the information of Album.
 - ✓ Shows the information and description of the Music, Performer.
 - ✓ To increase efficiency of managing the Music, Album
 - ✓ It deals with monitoring the information and transactions of Track.
 - ✓ Manage the information of Music.

1.4 TECHNOLOGIES

CSS (FRONT END)

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External stylesheets are stored in CSS files

Java Script (J.S)

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved.

- It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.

PHP

- PHP (Hypertext Preprocessor).
- PHP is an acronym for "PHP: Hypertext Preprocessor".
- PHP is a widely-used, open-source scripting language.
- PHP scripts are executed on the server.
- PHP is free to download and use.

MY SQL (DATABASE)

- MySQL is one of the most recognizable technologies in the modern big data ecosystem.
- Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it's clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL.

MUCIX(Website): -196400307098, 196400307100, 196400307105, 196400307112

- With MySQL, even those new to relational systems can immediately build fast, powerful, and secure data storage systems.
- MySQL's programmatic syntax and interfaces are also perfect gateways into the wide world of other popular query languages and structured data stores.

APACHE

- Apache HTTP server is world's most popular HTTP web server.
- Its fast and secure, and runs over half of all web server around the globe.
- Apache is also a free software. Apache web server has full range of features, including CGI (Common Gateway Interface), SSL (Secure Socket Layer) and virtual domains.
- It's also available for other kind of operating system like MacOS x, Windows.

BOOT STRAP

- Bootstrap is a free and open-source front end web framework for designing websites and web applications.
- It contains HTML a CSS base templates for typography, forms, buttons, navigation and other interface components.
- It is very useful in front end development and other designing development.

CH:2

PROJECT

MANAGEMENT

2.1 Project Planning and Approach

2.1.1 Project planning and Approach

2.2 Project Scheduling

2.3 Roles and Responsibility

2.4 Project Development Approach

2.4.1 Iterative Model

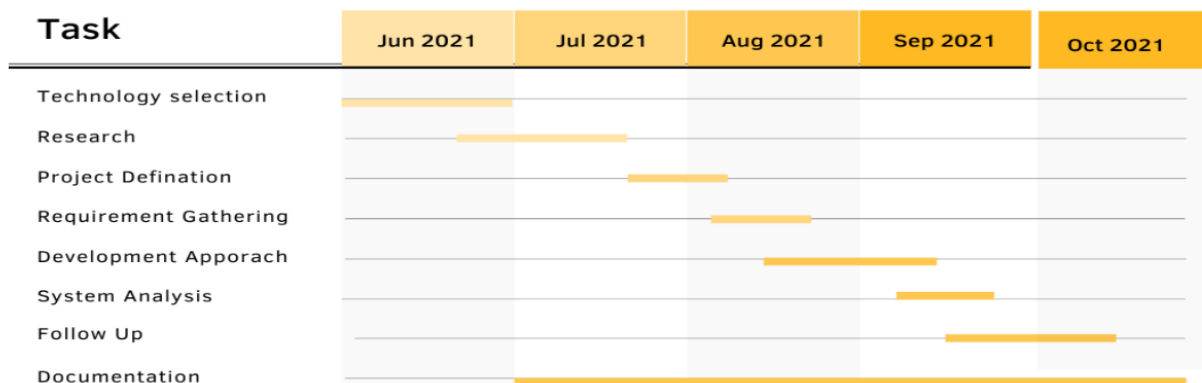
2.1 PROJECT PLANNING

2.1.1 PROJECT PLANNING AND APPROACH

- **Project Planning:** - Project planning is one of the major tasks which are performed during the development of the project. Using project, the task of finding the size of the project is done and with that total amount of time and cost required for project development is calculated. Planning of this project was done using a special approach after getting the project definition, upper-level analysis was performed first. That analysis was confined to the whole project level work. The analysis gave the idea about the size and structure of the project and using that analysis information, planning of the project was done.
- **PROJECT APPROACH:** The approach to develop the software the system should follow some systematic way i.e., Software development life cycle. Using the upper-level analysis and the environment of the project, which life cycle model would fit properly for this project was judged. After deciding the proper software development life cycle model, the development of this project according to the model was done.

2.2 Project Scheduling

- Project scheduling consists of identifying the tasks needed to complete the project, determine the dependency among different tasks plan the starting and depending dates for various tasks and determine the chain of tasks what determines the duration of project scheduling we decide the order in which to do the tasks.



2.3 Role And Responsibility

ACTIVITY	ROLE AND RESPONSIBILITY
Requirement Gathering	Samarth and Khush
Analysis	Manan and Kalp
Design	Kalp and manan
ER and Flowchart and DFD	Samarth and khush
Documentation	Samarth and Kalp

2.4 Project Development Approach

- This project follows the Iterative model for project development.

2.4.1 Iterative Model

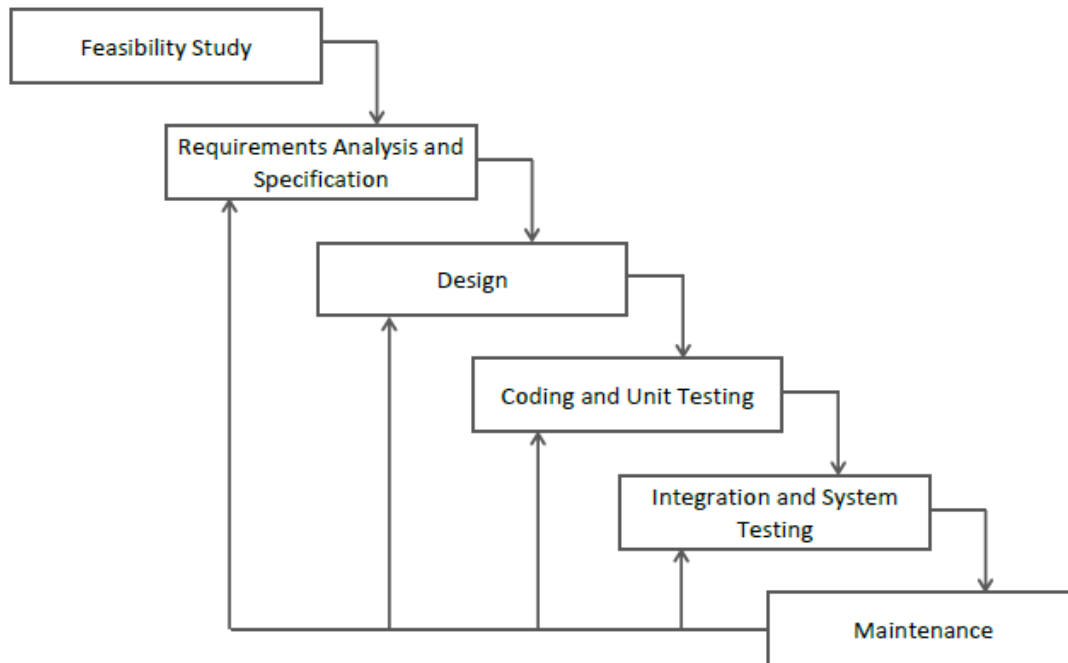
- The approach is in four steps: design, implement, test and review
- The translation of strategic and organizational goal into initiatives.
- The key to successful project is in the planning. Creating a project plan is the first thing you should do understanding any project.
- On completion of this guide, you should have project planning approach that you can use for future projects phases.

Phases of Iterative Waterfall Model

1. Requirement Analysis
2. Feasibility Study
3. Software Design
4. Coding/Implementation
5. Software Testing
6. Software Deployment
7. Software Maintenance

- We are choosing this model because this model is very easy to understand.
- This initiative water falls model's every phase contains feedback path to its previous phase.

- If any interrupt occurs in any phase of model, then it is simple to make changes and modification at any phase.
- By using this model, we can complete project earlier.
- By using this model, we have benefitted that customer involvement is not required during the software development.
- This model is suitable for large and complex projects.



CH:3

SYSTEM REQUIREMENT STUDY

3.1 Proposed System

3.1.1 Modules & Features in the New System

3.1.2 User Characteristics

3.1.3 Hardware & Software Requirements

3.1.4 Assumptions and Dependencies

3.1 PROPOSED SYSTEM

3.1.1 MODULES AND FEATURES IN THE NEW SYSTEM:

❖ MODULES:

- User:
 - ✓ Registration: user can register all the information.
 - ✓ Login: user can login in the system.
- Admin:
 - ✓ Update the details on the website which is necessary

3.1.2 USER CHARACTERISTICS:

❖ There are two types of user are as follow:

1. Admin: Administrative module is provided for the administrator to manage the site and update the content at the regular interval, the major operations included in this module are:
 - i. Handles all the users on the website.
 - ii. Update the details on the website which is necessary
2. User: This module for users. Using this modules user can take advantage of various function listed below:
 - i. Can sign-in for the registration.
 - ii. User can give feedback about the website.

3.1.3 HARDWARE & SOFTWARE REQUIREMENTS:

❖ SOFTWARE REQUIREMENTS:

- **SERVER-SIDE REQUIREMENT:**

- ✓ Operating System: Any windows OS.
- ✓ User Interface: HTML, CSS, J.S.
- ✓ Programming language: PHP.
- ✓ IDE/Workbench: Visual studio code.
- ✓ Database: MYSQL.

- **CLIENT-SIDE REQUIREMENT:**

- ✓ Operating System: Any Window OS.
- ✓ Browser: MYSQL Any Browser.

- **HARDWARE REQUIRMENT:**

- ✓ **SERVER-SIDE REQUIREMENT:**
- ✓ Processor: Intel Core i3.
- ✓ Hard Disk:40gb.
- ✓ RAM:4gb.

- **CLIENT-SIDE REQUIREMENT:**

- ✓ Processor: Pentium (Any Generation).
- ✓ Ram:1gb.

3.1.4 ASSUMPTIONS & DEPENDENCIES:

- ✓ There is no assumption as of now. All users of the system have different privileges.
- ✓ It is depended to web-based application like Internet Explorer and Mozilla Firefox.
- ✓ Administrator is created in the system already.
- ✓ Roles and tasks are predefined.
- ✓ This project is stand-alone project so it will not affect the system where it will not embed.
- ✓ This system will not depend on any other module.
- ✓ It is a web-based so every-one will independently use it.
- ✓ It will not affect the environment at all.

CH:4

SYSTEM ANALYSIS

4.1 Technically Feasibility Study

4.2 Feasibility Study

4.2.1 Economic Feasibility Study

4.2.2 Technical Feasibility Study

4.2.3 Operational Feasibility Study

4.3 Hardware Study

4.4 Identify Customer's needs.

4.5 Perform Economical and Technical Analysis.

4.6 Use Case Diagram

4.1 Technically Feasibility Study

- It includes the study of function, performance and constraints that may affect the ability to achieve an acceptable system.
- It is a measure of the practicality of a specific technical solution and the availability of technical resources and expertise of
- The proposed system uses html, CSS, Java Script as front-end and PHP as back-end tool.
- Oracle is a popular tool used to design and develop database objects such as table views, indexes.

4.2 Feasibility study

- It includes study and analyzing all the existing or required functionalities of the system.
- Feasibility study includes consideration of all the possible ways to provide a solution to the given problem.
- The proposed solution should satisfy all user requirement and should be flexible enough so that future change can be done easily.
- There are three types,
 - ✓ Economical Feasibility
 - ✓ Technical Feasibility
 - ✓ Operational Feasibility
- ✓ **Economical Feasibility:** - This study is carried out to check the economic impact which the system will have on the organization. The amount of fund which company can use in the research & development of the system is limited. The expenditure must be justified. Thus, the developed system is within a budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

- ✓ **Technical Feasibility:** - This study is carried out to check the feasibility which is the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This developed system must have a modest requirement, as only minimal or null change are required for implementing this system
- ✓ **Operational Feasibility:** - The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as necessity.

4.3 Hardware Study

Name of Component	Minimum Specification
Processor	Pentium 630MHz
Ram	Minimum 256 Mb
Hard disk	20 GB
Operating system	Windows and Linux

4.4 CUSTOMER NEEDS

- This guide outlines how to identify which of your customers are the most valuable to you. It also provides tips on selling more to them and attracting new high-value customers.
- The benefits of understanding your customers are follow: -
 - ✓ Learn about your customers
 - ✓ Make customer information available
 - ✓ Analyze your customers
 - ✓ What makes your customers valuable?
 - ✓ Enhance the customer experience

- ✓ Market more effectively
- ✓ Find new customers

4.4.1 Learn about your customers

- Your customers are a valuable source of information, so you should aim to collect data that lets you identify your customers and how they behave.
- This will vary depending on your customer profile. If you sell to individual consumers, you might want to know about their age, gender, income and so on.
- For businesses, you might want to know what industry they operate in and their size.

4.4.2 Analyze your customers

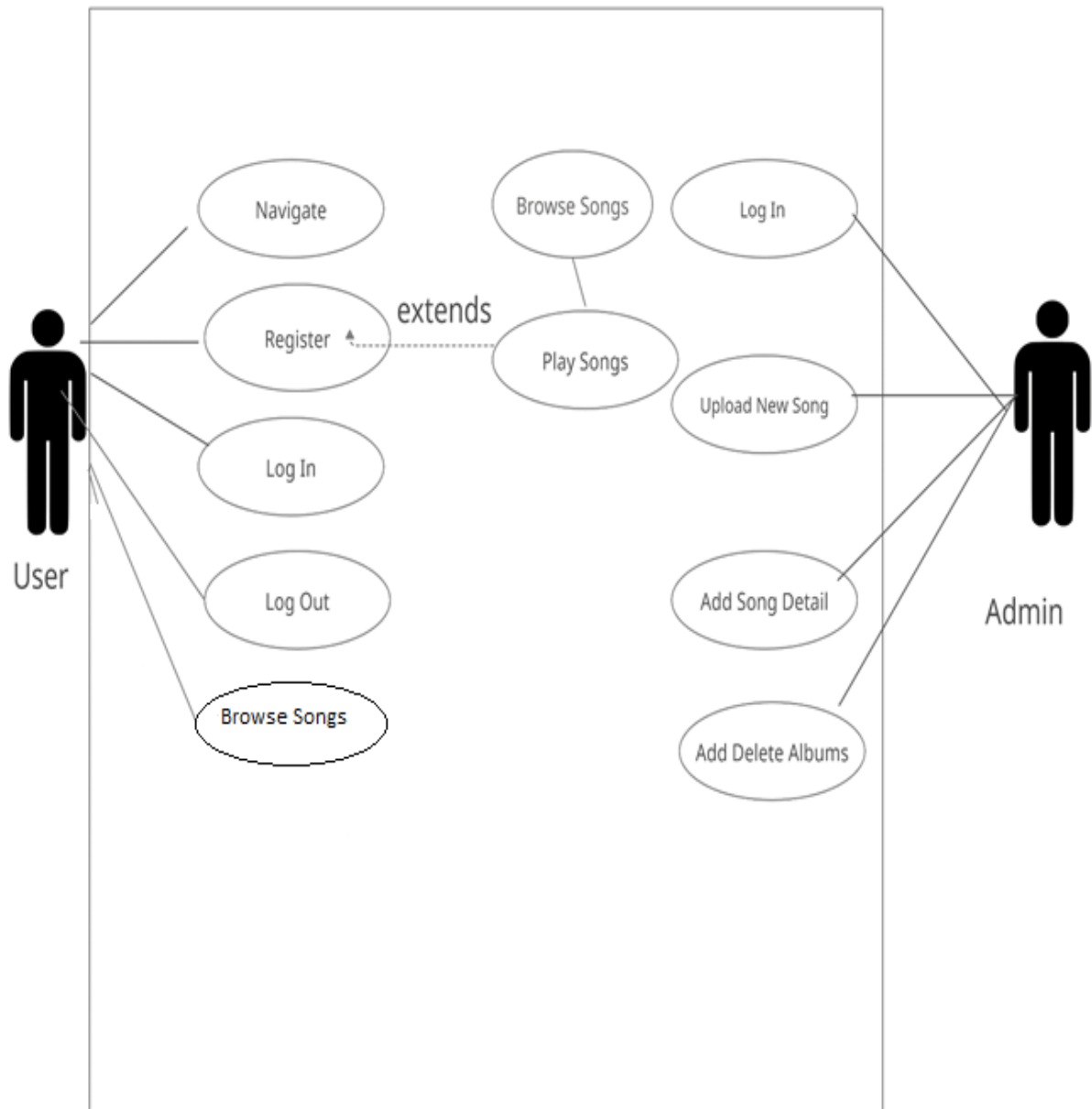
The right information will let you build up a useful profile of your customers. This typically includes the following:

- who they are - the age and gender of individual consumers, or industry and business size for corporate customers?
- what they think and believe, what interests them and their opinion of you and your product
- their purchasing behavior - which products they buy, where they buy them, when, and how they pay

4.5 ECONOMIC ANALYSIS

- Economic analysis of projects helps identify and select public investments that will sustainably improve the welfare of beneficiaries and a country as a whole.
- This 2nd edition pamphlet¹ outlines key areas of economic analysis of projects.
- It stresses that analysis begins during country strategy studies and programming, when projects are identified, and continues iteratively throughout the project cycle.
- Economic analysis is coordinated with institutional, financial, environmental, social, and poverty analyses, forming an integral part of investment appraisal.

4.6 USE CASE Diagram



CH:5

SYSTEM

DESIGN

5.1 Database Design/Data Structure Design

5.1.1 Data Dictionary

5.1.2 ER Diagram

5.1.3 Data Flow Diagram

5.2 Input/output and Interface Design

5.2.1 Flow Chart

5.1 DATABASE DESIGN/DATA STRUCTURE DESIGN

5.1.1 DATA DICTIONARY

USERS:

Field Name	Data Type	Field Size	Constraint
id	Int	11	Primary Key
Username	Varchar	25	Not Null
Firstname	Varchar	50	Not Null
Lastname	Varchar	50	Not Null
Email	Varchar	55	Unique
Password	Varchar	32	Not Null
Signupdate	DateTIme	30	Not Null
Profilepic	Varchar	500	Not Null

SONGS:

Field Name	Data Type	Field Size	Constraint
id	Int	11	Primary Key
title	Varchar	50	Not Null
artist	Int	11	Not Null
album	Int	11	Not Null
genre	Int	11	Not Null
duration	Varchar	8	Not Null
path	Int	55	Not Null
albumOrder	Int	11	Not Null
plays	Int	11	Not Null

ARTISTS:

Field Name	Data Type	Field Size	Constraint
Id	Int	11	Primary Key
Name	Varchar	50	Not Null

ALBUMS:

Field Name	Data Type	Field Size	Constraint
Id	Int	11	Primary Key
Name	Varchar	50	Not Null

GENRES:

Field Name	Data Type	Field Size	Constraint
Id	Int	11	Primary Key
Title	Varchar	50	Not Null
Artist	Int	11	Not Null
Genre	Int	11	Not Null
ArtworkPath	Varchar	255	Not Null

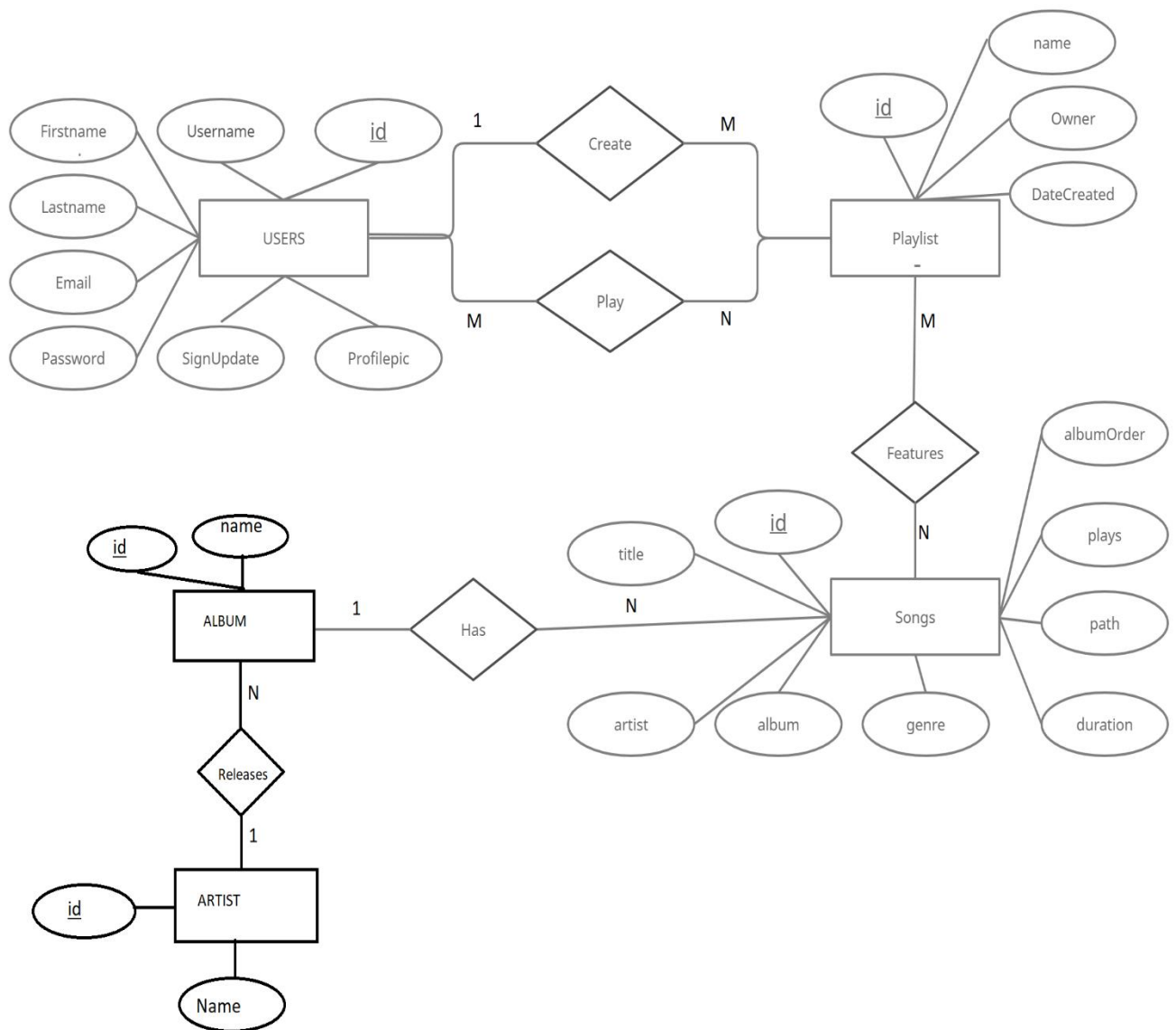
PLAYLIST:

Field Name	Data Type	Field Size	Constraint
Id	Int	11	Primary Key
Name	Varchar	50	Not Null
Owner	Varchar	50	Not Null
dataCreated	datetime		Not Null

PLAYLISTSONGS:

Field Name	Data Type	Field Size	Constraint
Id	int	11	Primary Key
Songid	Int	11	Not Null
playlistid	Int	11	Not Null
playlistOrder	Int	11	Not Null

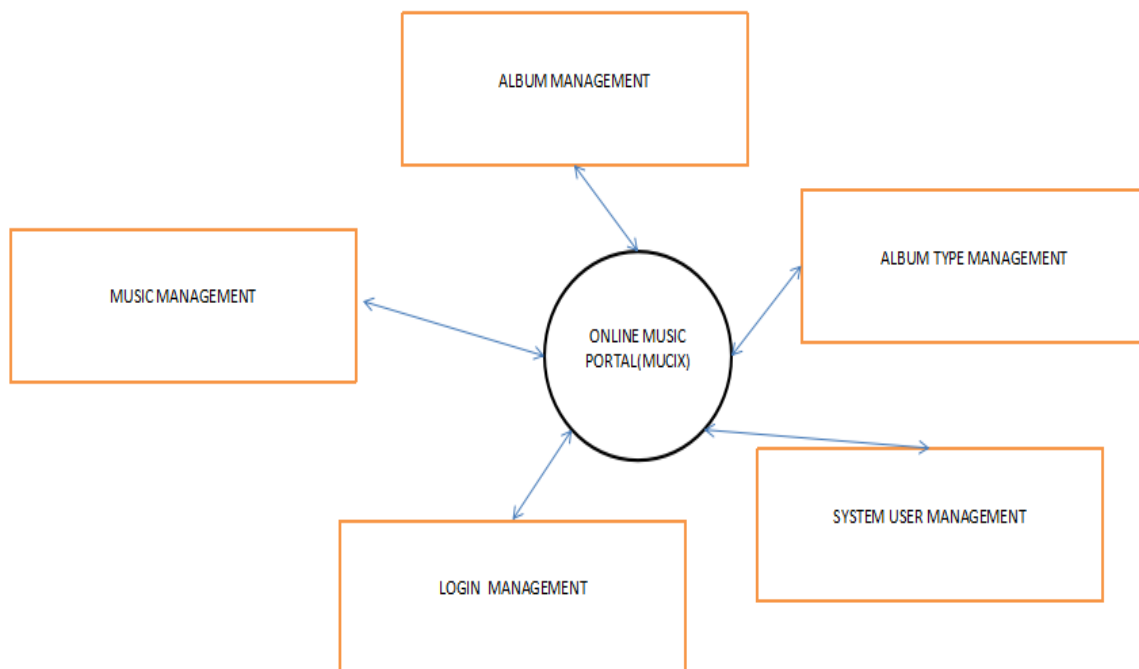
5.1.2 E. R DIAGRAM



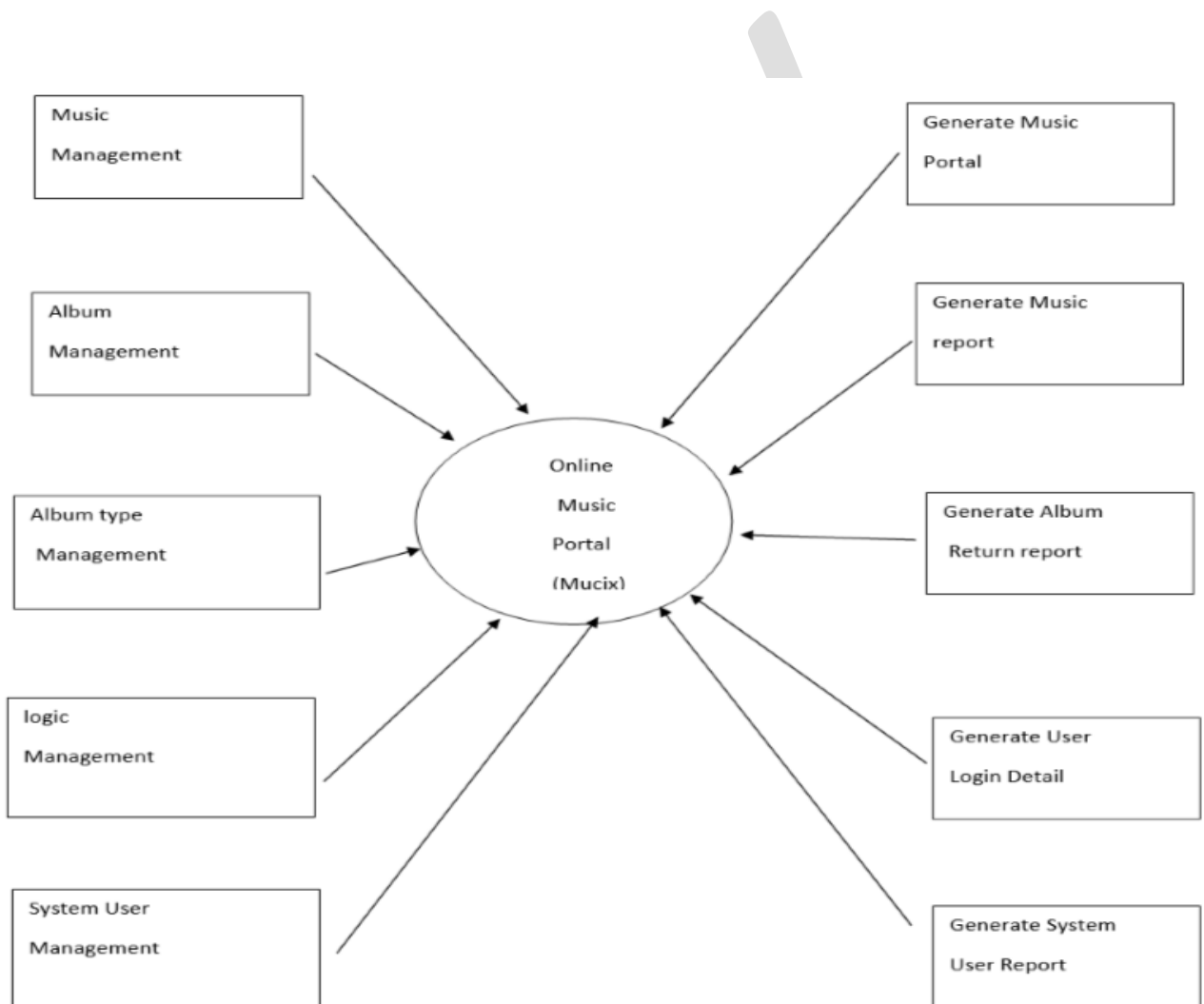
5.1.3 DFD (Data Flow Diagram)

- ✓ A data flow diagram (DFD) maps out the flow of information for any process or system.
- ✓ It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.
- ✓ Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

- **DFD LEVEL 0(Data Flow Diagram)**

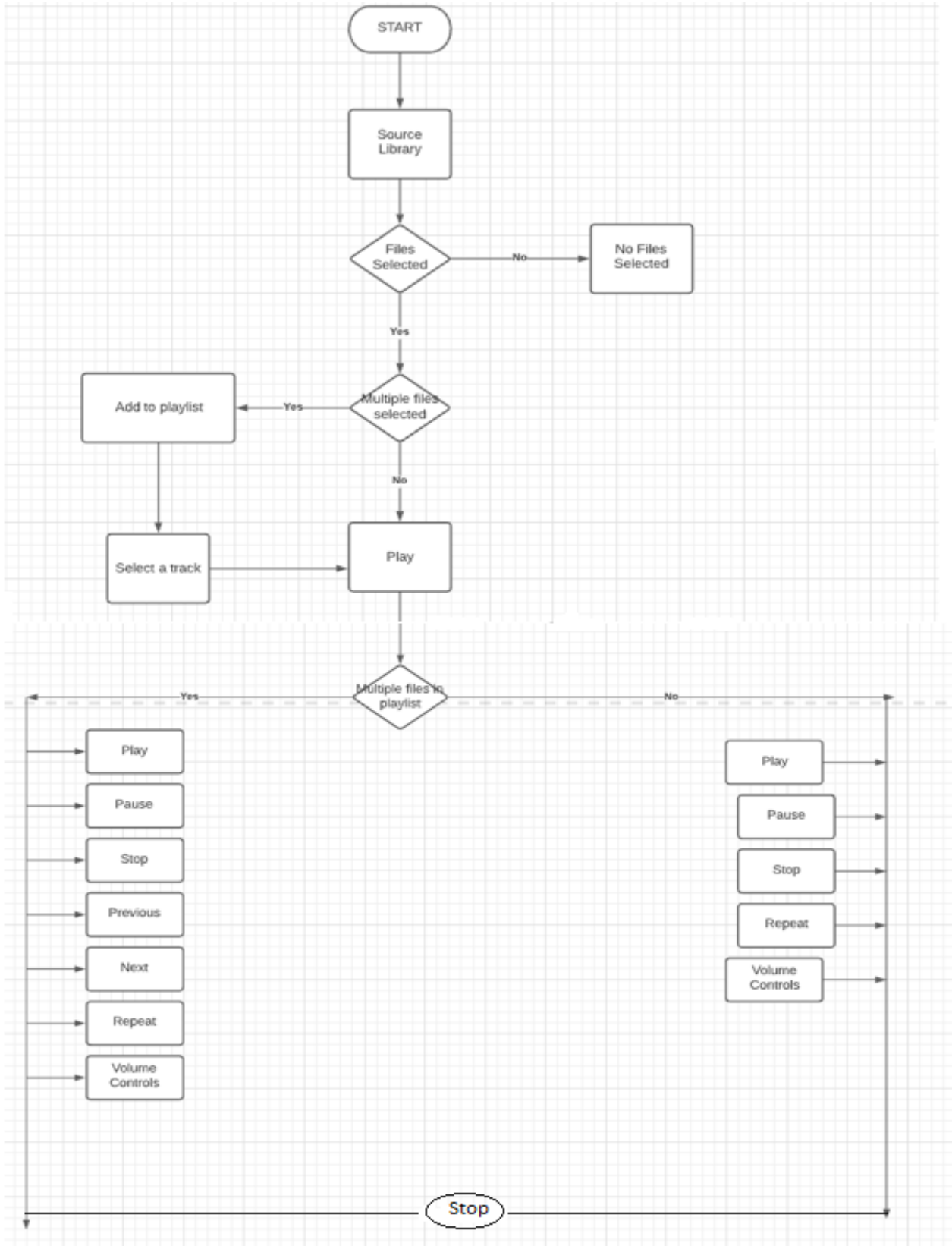


- **DFD LEVEL 1 (Data Flow Diagram)**



5.2 INPUT / OUTPUT AND INTERFACE DESIGN

5.2.1 FLOW CHART



CH:6

IMPLEMENTATION

6.1 Implementation Environment

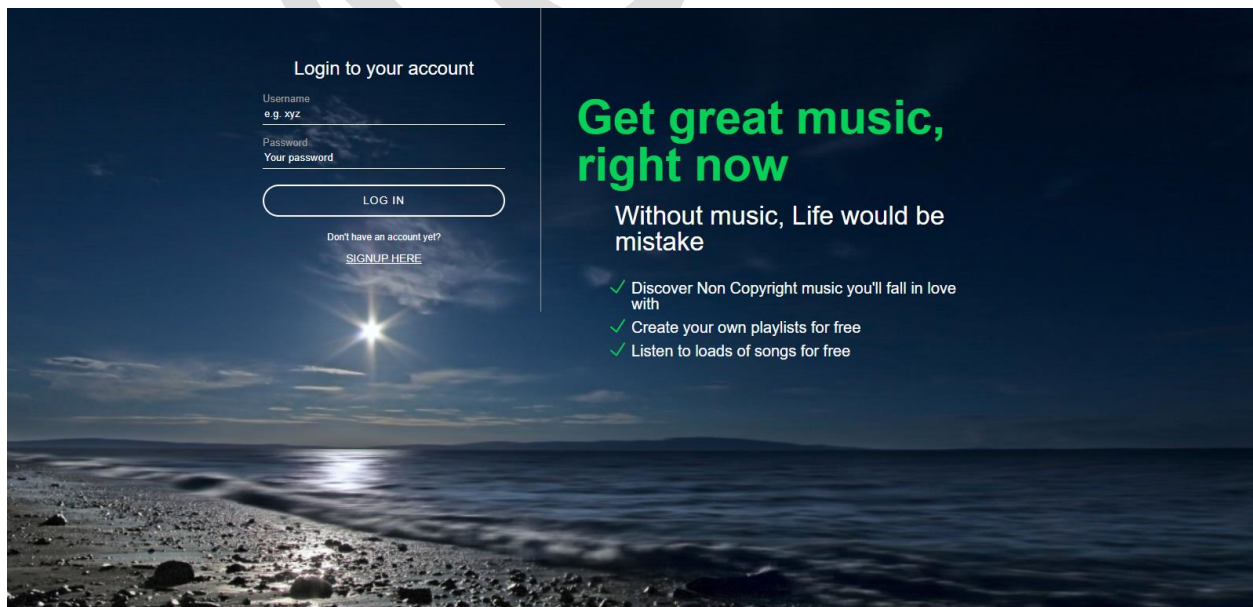
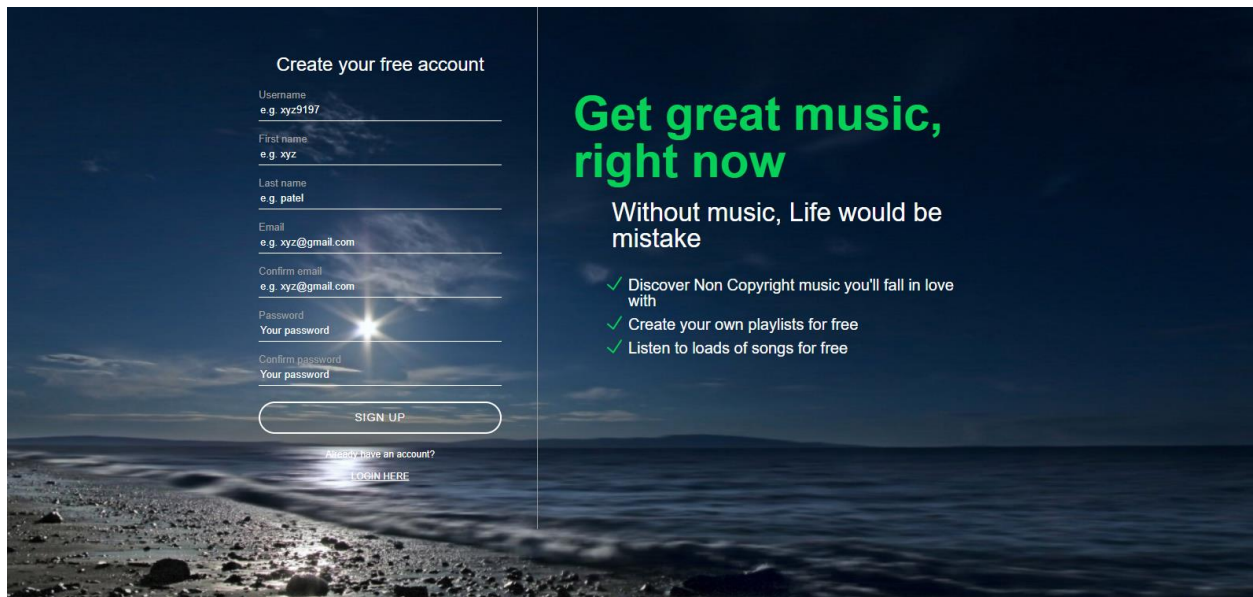
6.2 Screenshots of Forms, Reports and Interface

6.3 Sample Coding

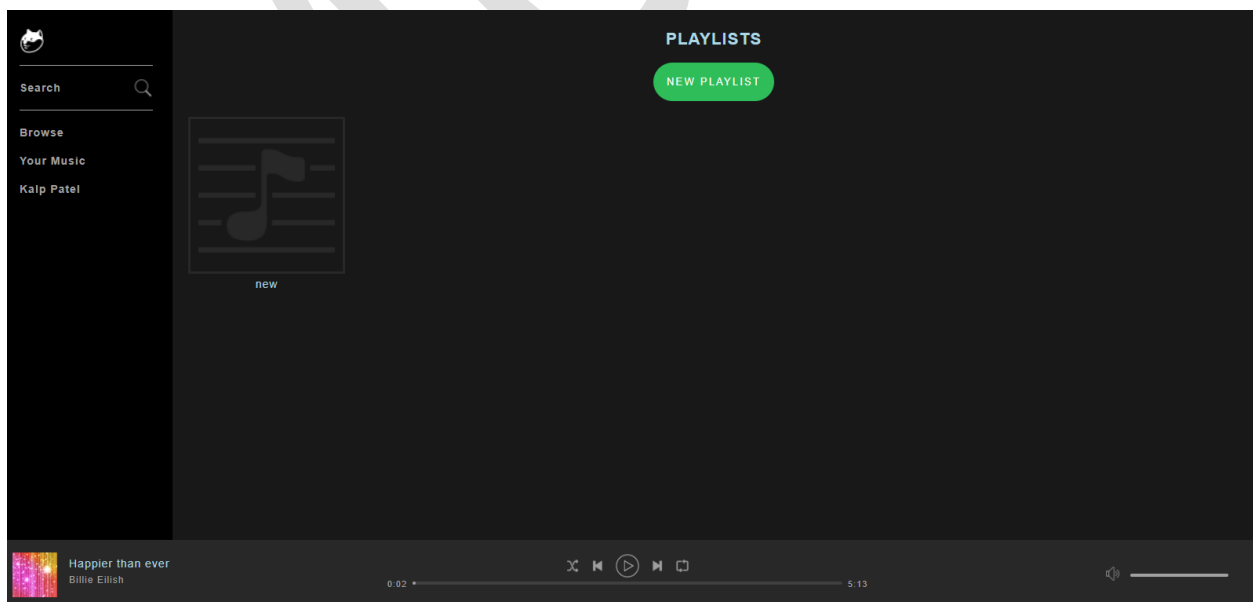
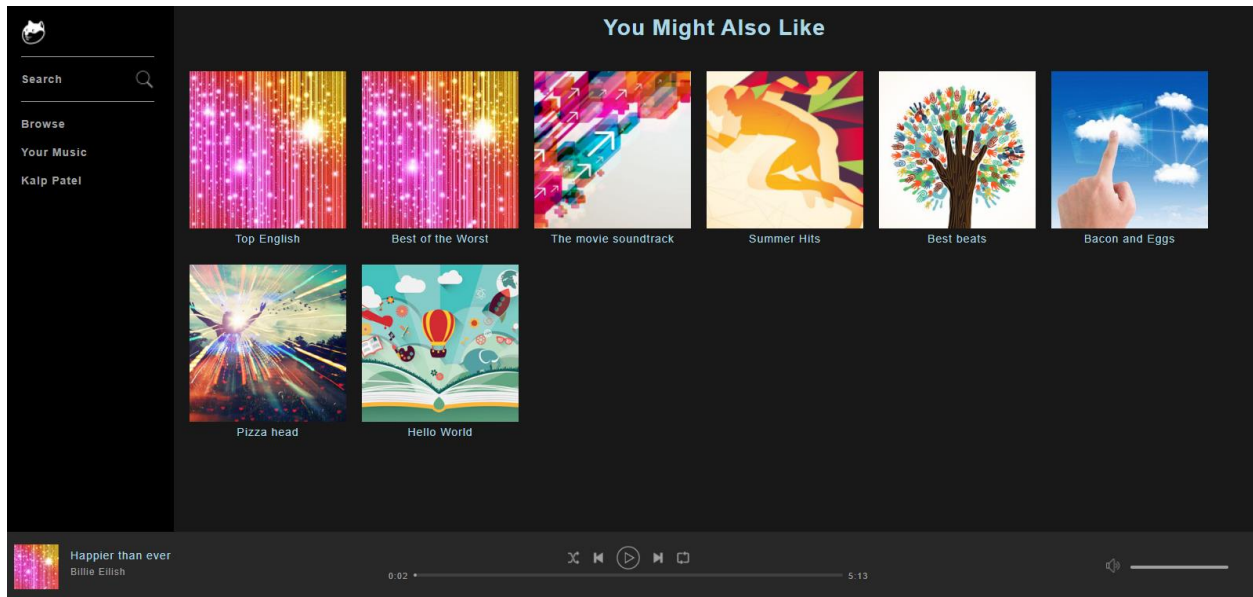
6.1 Implementation Environment

In Online Music Portal System (MuCiX) , there are two users, Main user and Admin. This paper gives our researching work on how to design a music player based on Online Music Portal System. The music player, which uses the front-back-end architecture, is divided into the part of music playback and the part of player interface and music list. Besides the playback functions, the music player can also do music rating by user preferences, showing the previous and next music name with corresponding information and operating controller into a panel. The music player runs stably and conveniently during testing. Data of users is saved in My Sql Database, so that main user can access it when needed.

6.2 SCREENSHOT



MUCIX(Website): -196400307098, 196400307100, 196400307105, 196400307112



RCTI(CE)

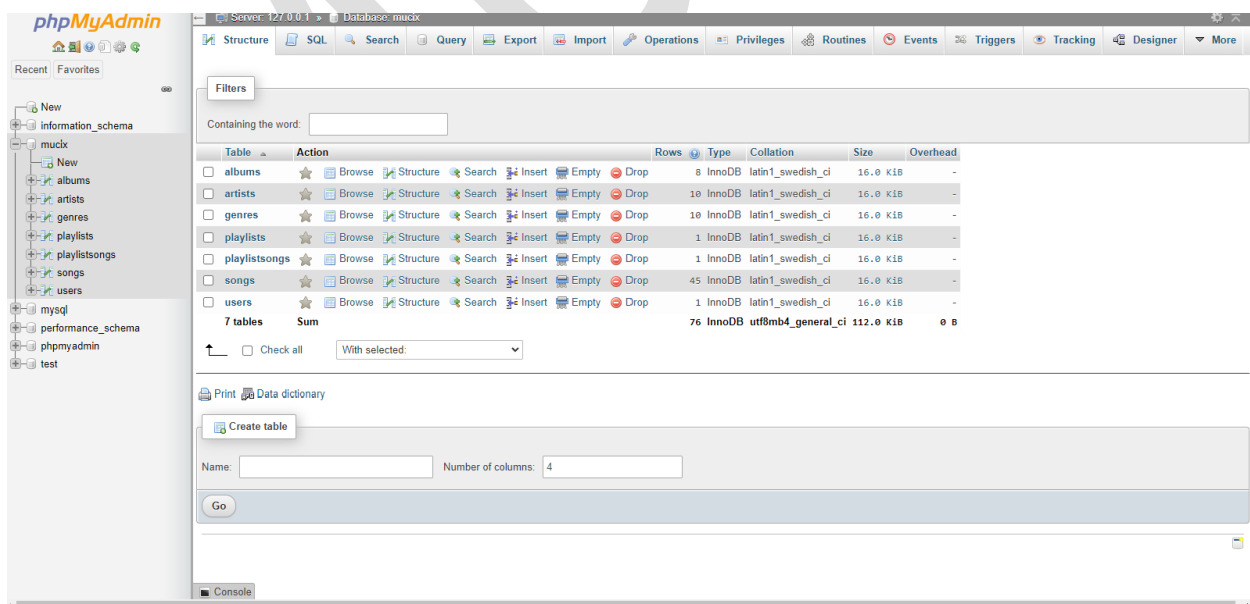
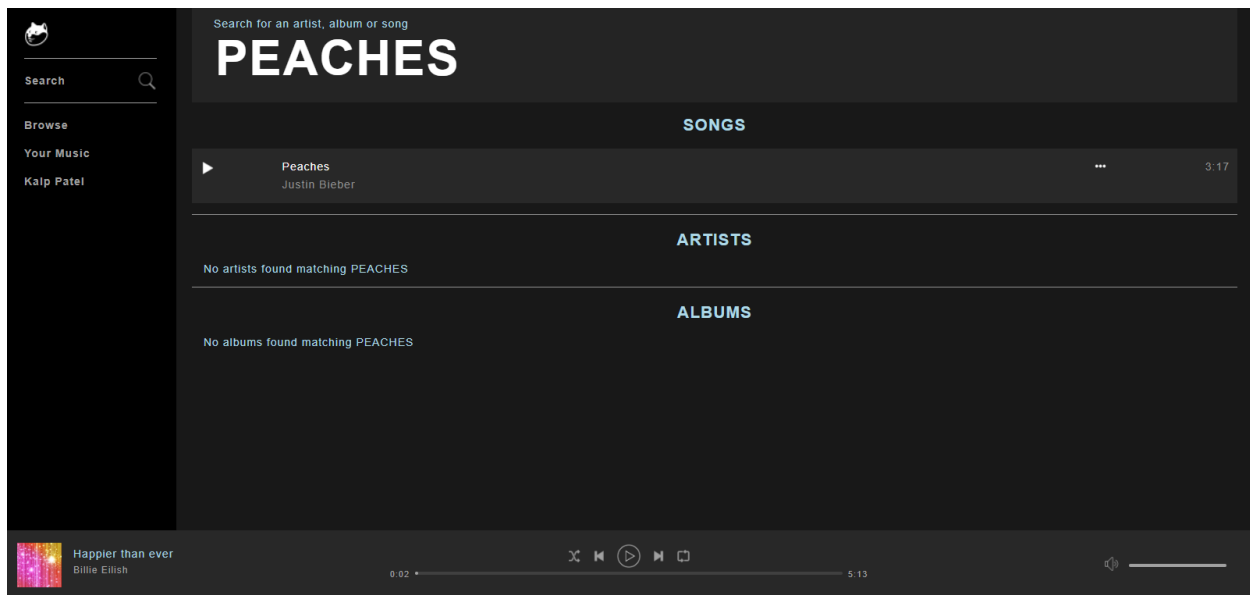
MUCIX(Website): -196400307098, 196400307100, 196400307105, 196400307112

The screenshot displays the MUCIX website's registration interface. On the left is a dark sidebar with a cat icon, a search bar, and links for 'Browse', 'Your Music', and 'Kalp Patel'. The main content area has a dark background. The 'EMAIL' section contains a text input field with 'Pkalp338@gmail.com' and a 'SAVE' button. Below this, the 'PASSWORD' section has three text input fields labeled 'Current password', 'New password', and 'Confirm password', followed by another 'SAVE' button. At the bottom, a music player shows 'Happier than ever' by Billie Eilish with a progress bar from 0:02 to 5:13.

This screenshot shows the user profile page for 'Kalp Patel' on the MUCIX website. The sidebar is identical to the previous one. The main content area features the name 'Kalp Patel' at the top, followed by two buttons: 'USER DETAILS' and 'LOGOUT'. The bottom music player remains the same, displaying 'Happier than ever' by Billie Eilish.

RCTI(CE)

MUCIX(Website): -196400307098, 196400307100, 196400307105, 196400307112



RCTI(CE)

6.3 SAMPLE CODING

1. Register.php

```
<?php
    include("includes/config.php");
    include("includes/classes/Account.php");
    include("includes/classes/Constants.php");

    $account = new Account($con);

    include("includes/handlers/register-handler.php");
    include("includes/handlers/login-handler.php");

    function getInputValue($name) {
        if(isset($_POST[$name])) {
            echo $_POST[$name];
        }
    }
?>

<html>
<head>
    <title>Welcome to Mucix!</title>

    <link rel="stylesheet" type="text/css" href="assets/css/register.css">

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scri
pt>
    <script src="assets/js/register.js"></script>
</head>
<body>
    <?php
```

```

        if(isset($_POST['registerButton'])) {
            echo '<script>
                $(document).ready(function() {
                    $("#loginForm").hide();
                    $("#registerForm").show();
                });
            </script>';
        }
        else {
            echo '<script>
                $(document).ready(function() {
                    $("#loginForm").show();
                    $("#registerForm").hide();
                });
            </script>';
        }
    ?>

    <div id="background">

        <div id="loginContainer">

            <div id="inputContainer">
                <form id="loginForm" action="register.php"
method="POST">

                    <h2>Login to your account</h2>
                    <p>
                        <?php echo $account-
>getError(Constants::$loginFailed); ?>
                        <label
for="loginUsername">Username</label>
                        <input id="loginUsername"
name="loginUsername" type="text" placeholder="e.g Khush" value="<?php
getInputValue('loginUsername') ?>" required>
                    </p>

```

```
<p>
    <label
for="loginPassword">Password</label>
    <input id="loginPassword"
name="loginPassword" type="password" placeholder="*****"
required>
</p>

    <button type="submit"
name="loginButton">LOG IN</button>

    <div class="hasAccountText">
        <span id="hideLogin">Don't have an
account yet? Signup here.</span>
    </div>
</form>

<form id="registerForm" action="register.php"
method="POST">
    <h2>Create your free account</h2>
    <p>
        <?php echo $account-
>getError(Constants::$usernameCharacters); ?>
        <?php echo $account-
>getError(Constants::$usernameTaken); ?>
        <label
for="username">Username</label>
        <input id="username"
name="username" type="text" placeholder="e.g Khush998" value="<?php
getInputValue('username') ?>" required>
    </p>

    <p>
```

```
<?php echo $account-
>getError(Constants::$firstNameCharacters); ?>
<label for="firstName">First
name</label>

<input id="firstName"
name="firstName" type="text" placeholder="e.g. Khush" value="<?php
getInputValue('firstName') ?>" required>
</p>

<p>
<?php echo $account-
>getError(Constants::$lastNameCharacters); ?>
<label for="lastName">Last
name</label>

<input id="lastName"
name="lastName" type="text" placeholder="e.g. Patel" value="<?php
getInputValue('lastName') ?>" required>
</p>

<p>
<?php echo $account-
>getError(Constants::$emailsDoNotMatch); ?>
<?php echo $account-
>getError(Constants::$emailInvalid); ?>
<?php echo $account-
>getError(Constants::$emailTaken); ?>
<label for="email">Email</label>
<input id="email" name="email"
type="email" placeholder="e.g. example@gmail.com" value="<?php
getInputValue('email') ?>" required>
</p>

<p>
<label for="email2">Confirm
email</label>
```



```
<input id="email2" name="email2"
type="email" placeholder="e.g. example@gmail.com" value="<?php
getInputValue('email2') ?>" required>
</p>

<p>
    <?php echo $account-
>getError(Constants::$passwordsDoNoMatch); ?>
    <?php echo $account-
>getError(Constants::$passwordNotAlphanumeric); ?>
    <?php echo $account-
>getError(Constants::$passwordCharacters); ?>
    <label
for="password">Password</label>
    <input id="password"
name="password" type="password" placeholder="*****" required>
</p>

<p>
    <label for="password2">Confirm
password</label>
    <input id="password2"
name="password2" type="password" placeholder="*****" required>
</p>

    <button type="submit"
name="registerButton">SIGN UP</button>

    <div class="hasAccountText">
        <span id="hideRegister">Already
have an account? Log in here.</span>
    </div>

</form>

</div>
```

```
<div id="loginText">

    <h1> Get great music, right now
    </h1>

</div>

</div>

</div>

</body>
</html>
```

2. Index.php

```
<?php
include("includes/config.php");

//session_destroy(); LOGOUT

if(isset($_SESSION['userLoggedIn'])) {
    $userLoggedIn = $_SESSION['userLoggedIn'];
}
else {
    header("Location: register.php");
}

?>
```

```
<html>
<head>
  <title>Welcome to mucix</title>

  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
</head>

<body>

  <div id="nowPlayingBarContainer">

    <div id="nowPlayingBar">

      <div id="nowPlayingLeft">
        <div class="content">
          <span class="albumLink">
            
          </span>

          <div class="trackInfo">

            <span class="trackName">
              <span>Excuses</span>
            </span>

            <span class="artistName">
              <span>A.P Dhillon</span>
            </span>

          </div>

        </div>

      </div>

    </div>

  </div>
```

</div>

<div id="nowPlayingCenter">

<div class="content playerControls">

<div class="buttons">

<button class="controlButton shuffle"

title="Shuffle button">

</button>

<button class="controlButton

previous" title="Previous button">

</button>

<button class="controlButton play"

title="Play button">

</button>

<button class="controlButton pause"

title="Pause button" style="display: none;">

</button>

<button class="controlButton next"

title="Next button">

</button>

```

                                <button class="controlButton repeat"
title="Repeat button">
                                
                                </button>

                                </div>

                                <div class="playbackBar">
                                <span class="progressTime
current">0.00</span>
                                <div class="progressBar">
                                    <div class="progressBarBg">
                                        <div
class="progress"></div>
                                    </div>
                                </div>
                                <span class="progressTime
remaining">0.00</span>
                                </div>

                                </div>

                                </div>

                                <div id="nowPlayingRight">
                                    <div class="volumeBar">
```

```

                                <button class="controlButton volume"
title="Volume button">
                                
                                </button>

                                <div class="progressBar">
                                    <div class="progressBarBg">
                                        <div class="progress"></div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </body>
    </html>
```

CH:7

Testing

7.1 Testing Plan & Strategy

7.2 Testing Methods

7.3 Test Cases

7.0 Testing

Software testing is the process of evaluating and verifying that a website does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

7.1 Testing Plan And Strategy

Project Description: In Online Music Portal System (MUCIX) , we divided every functions in modules and tested separately. In CRUD operations we check every exception based on whether it was based on Internet Permission or Data Access Permission or if the same user is trying to get access different user data with limitations so that in database should not get overloaded at the same time.

7.2 Testing Methods

Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage.

System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

Performance Testing

Performance Testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into three types:

- **Load testing** is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it's designed to handle.
- **Stress testing** takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you'll be able to find the failure point of your piece of software.
- **Endurance testing**, also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.

Security Testing

With the rise of cloud-based testing platforms and cyber attacks, there is a growing concern and need for the security of data being used and stored in software. Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses. There are multiple types of this testing method, each of which aimed at verifying six basic principles of security:

1. Integrity
2. Confidentiality
3. Authentication
4. Authorization
5. Availability

6. Non-repudiation

7.3 TEST CASES

Login Page:

Test Case Id	Test Case Name	Test Case Description	Test Case steps	Expected	Actual
Login	Input Username	In Login form take correct username	Enter Username	Accept Username	User Navigated to the Password Field
	Input Password	In Login form take input username	Enter Password	Accept Password	User Focused to the Login Button
	Input Login Button Click	Validate Valid Username and Password	Click on Login Button	Accept Username and password	Go to the Home Page
		Validate Valid Username and Password	Click on Login Button	Not Accept Username and password	Reset Input field with error message
		Validate Valid Username and Password	Click on Login Button	Not Accept Username and password	Reset Input field with error message user already registered with different user type

Register Page:

Test Case Id	Test Case Name	Test Case Description	Test Case steps	Expected	Actual
Register	Input Username	In Register form take correct username	Enter Username	Accept Username	User Navigated to the First Name Field
	Input First Name	In Register form take correct First name	Enter First Name	Accept First Name	User Navigated to the last Name Field
	Input last Name	In Register form take correct last name	Enter last Name	Accept last Name	User Navigated to the Email Field
	Input Email	In Register form take correct Email	Enter Email	Accept Email	User Navigated to the Confirm Email Field
	Input Confirm Email	In Register form take correct Confirm Email	Enter Confirm Email	Accept Confirm Email	User Navigated to the Password Field
	Input Password	In Register form take correct Password	Enter Password	Accept Password	User Navigated to the Confirm Password Field
	Input Password	In Register form take correct Confirm Password	Enter Confirm Password	Accept Confirm Password	User focused to the Sign Up Button
	Input Sign Up Button Click	Validate Valid User name, first name, last name, email, confirm email, Password	Click on Sign up Button	Accept User name, first name, last name, email, confirm email,	Go to the Home Page

		and confirm Password		Password and confirm Password	
--	--	-------------------------	--	-------------------------------------	--

CH:8

SECURITY MEASURES AND FUTURE ENHANCEMENT

SECURITY MEASURES:

- The security measures imposed in the software are: -
 1. A login password is provided in the software. User must login to activate the application.
 2. User cannot change the password. To change password, he must contact the administrator.
 3. The user/password are given through SQL.
 4. Data security, correctness integrity is checked up before saving, update or delete if errors found the procedure is aborted.
 5. A primary key & foreign key concept is implemented for avoiding incorrect data entry or intentional or accidental delete or modification of data.
 6. When user tries to delete the data then this first check for its reference used by other data, if found the deletion aborted.
 7. We are also providing various securities at user level or at forms.

FUTURE ENHANCEMENT:

- On the basis of the work done in dissertation entitled “MuCiX”, the following conclusions emerge from the development:

1. This project has achieved the objective of replacing/augmenting the conventional system.
2. The development of this package has been achieved by using HTML, C.S.S, J.S, which is very conducive to develop the package with regard to time and specific need to the user.
3. This package is highly user friendly, required an optimal minimal input from user while providing highly relevant and focused outputs.
4. Fully automated, avoiding human intervention. Hence it provides a very rapid cost effective alternative to the conventional manual operation/procedures; the visual outputs are more reliable than the audio forms of manual communication.
5. The system can further extend as per user and administrative requirements.

CH:9

CONCLUSION/SUM MARY

CONCLUSION / SUMMARY:

- Instead of downloading multiple apps and wasting all storage space, this proposal reduces the use of a lot of applications and will be a complete music solution.
- Streaming music, or more accurately streaming audio, is a way of delivering sound — including music — without requiring you to download files from the internet. Music services like Spotify, Pandora, and Apple Music use this method to provide songs that can be enjoyed on all types of devices.
- It tracks all the information of Performer, Track, Customer act Manage the information of Performer.
- It Shows the information and description of the Music, Album.
- In this music website we can play songs for free.
- It provides the functionality like creating a playlist and also has search functionality which will the search the song and give the desired result. It provides free no copyright music lists below and browse high quality, free non-copyrighted music by top artists.
- Protecting the privacy and security of your information is our priority for everyone at “Mucix”.
- It also provides the variety of non-copyright songs.

CH:10

REFERENCES/BIBIBI

OGRAPHY

<https://codewithharry.com/videos/php-tutorials-in-hindi-1/>

<https://codewithharry.com/videos/javascript-tutorials-in-hindi-1/>

<https://www.w3schools.com/php/default.asp>

https://www.w3schools.com/css/css_selectors.asp

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

<https://mdbootstrap.com/docs/b4/jquery/forms/search/#:~:text=Bootstrap%20search%20is%20a%20component,Databases>

https://en.wikipedia.org/wiki/Feasibility_study