CHAPTER 1
INTRODUCTION

1.Introduction

The **“RYTHM RISE”** project is a web-based application developed to showcase fundamental web development skills using HTML and CSS. The project aims to create a simple yet visually engaging music website with a clean layout and basic interactive functionality. With the increasing demand for user-friendly and aesthetically pleasing websites, this project demonstrates how HTML and CSS can be effectively utilized to build a functional and stylish interface. The website features a minimalistic design with clearly defined sections, including a title, interactive buttons, and background styling. The HTML component defines the webpage’s structure, organizing the content into headings, buttons, and layout sections. The CSS component is used to enhance the visual appearance by adding colors, padding, margins, and hover effects, making the website dynamic and user-friendly. The primary objective of this project is to develop a static website that demonstrates clean code structure, responsive design, and basic interactivity. The website is designed to be easily navigable and visually appealing, offering a practical demonstration of how front-end technologies can be combined to create a polished and professional-looking web interface. By developing this project, the team gained practical experience in designing and styling web pages, applying CSS rules, and ensuring cross-browser compatibility. The “RYTHM RISE” project not only serves as a practical demonstration of front-end web development but also provides a strong foundation for future enhancements, such as adding audio playback functionality, navigation menus, and animations.

1.1Problem definition

In the current digital landscape, creating visually appealing and user-friendly websites is essential for providing a seamless browsing experience. However, many beginner-level projects lack structured design and consistent styling. This project addresses the need for a simple yet elegant music website that demonstrates clean web design practices using HTML and CSS. The goal is to build a static website with a clear layout, interactive buttons, and basic styling, offering a practical example of front-end web development skills.

1.2 Existing System

Traditional music websites often feature complex structures with extensive functionality, such as user authentication, music streaming, dynamic content management, and advanced search filters. These systems typically require backend servers, databases, and third-party APIs to manage large-scale content and user interactions. While such platforms offer robust features, they involve heavy resource usage, complex coding, and ongoing maintenance, making them difficult for beginners to develop and manage. In contrast, the "RYTHM RISE" project adopts a minimalist design with basic interactivity, focusing solely on frontend development using HTML and CSS. This approach simplifies the website's structure, making it lightweight, easy to build, and highly accessible for learners. Despite its simplicity, the project delivers a visually appealing and functional interface with interactive buttons and clean styling, offering a practical demonstration of basic web development concepts.

1.3 Proposed System

The proposed system aims to develop a music website with a clean, responsive, and user-friendly interface. It focuses on providing a visually appealing design with clearly defined sections, including a title, buttons, and background styling. The website is built entirely using HTML and CSS, making it lightweight and easy to deploy. The HTML defines the structure, organizing content into logical sections, while CSS enhances the appearance with colours, padding, and hover effects, creating a more engaging user experience. The website also includes interactive buttons that simulate basic interactivity, such as potential music playback functionality in the future. Its responsive design ensures consistent rendering across different devices and screen sizes, offering a seamless experience. This simplified yet scalable approach makes the project highly accessible for learners while providing a solid foundation for future enhancements, such as integrating audio players, animations, or expanding the styling effect.

1.4 Literature review

Literature Survey for Music Player Application

Music player applications have significantly evolved from traditional media playback devices to modern, feature-rich software platforms. Early applications like Winamp and Windows Media Player focused on basic playback functionalities, whereas contemporary platforms such as Spotify and Apple Music offer advanced capabilities, including cloud storage, AI-driven recommendations, and streaming services. The transition from offline to online music consumption has transformed how users interact with their music libraries, providing seamless accessibility across multiple devices. Additionally, modern music players support a wide range of audio and video formats like MP3, WAV, AAC, MP4, and MKV, ensuring compatibility with diverse media content. Essential features such as playlist management, equalizers, and custom sound enhancements allow users to tailor their listening experience, while cloud integration enables collaborative playlist sharing and real-time synchronization.

The development of music player applications involves various technologies and challenges. Popular programming languages include Java, Kotlin for Android, and Swift for iOS, with frameworks like Flutter and React Native facilitating cross-platform compatibility. Media libraries such as ExoPlayer, AV Foundation, and VLC framework provide robust playback support, while APIs like Spotify API and YouTube API enhance streaming capabilities. However, challenges such as copyright and licensing compliance, efficient resource management for handling large media libraries, and AI-based personalization persist in the industry. Future trends indicate advancements in AI-powered music curation, voice assistant integration, and immersive audio technologies such as 3D and spatial audio, further enhancing the listening experience.

CHAPTER 2

SYSTEM REQUIREMENTS

2.System Requirements

The “RYTHM RISE” project requires a standard computer or laptop with a web browser (e.g., Chrome, Firefox) and a text editor (e.g., VS Code) for coding. No additional software or backend services are needed.

2.1 Hardware & Software Requirements

The “RYTHM RISE” project requires a standard computer or laptop with at least 4 GB of RAM and a dual-core processor to ensure smooth performance during development and testing. A stable internet connection is recommended for accessing external resources, downloading necessary tools, and testing the website in different environments. Since the project is entirely front-end based, it does not demand high-end hardware or large storage capacity, making it accessible on most modern computers.

On the software side, the project uses a web browser, such as Chrome, Firefox, or Edge, to display and test the website. A text editor like Visual Studio Code (VS Code), Notepad++, or Sublime Text is required for writing and editing the HTML and CSS code. These editors offer helpful features like syntax highlighting and live previews, enhancing the development process. The project is compatible with popular operating systems, including Windows, macOS, and Linux, ensuring it can be developed and tested across different platforms. No additional software installations or complex configurations are necessary, making the project easy to set up and execute.

2.2 Software Requirements specification (SRS)

The “RYTHM RISE” project follows a simple and clear Software Requirements Specification (SRS) to outline the necessary components, functionalities, and objectives of the system. The primary goal is to create a static music website using HTML and CSS, focusing on a clean design and basic interactivity.

The functional requirements include the ability to display a visually appealing layout with buttons that offer basic interaction. The website should render properly on different browsers and devices, ensuring a consistent user experience. Non-functional requirements include fast loading time, compatibility with multiple operating systems (Windows, macOS, and Linux), and ease of use, making it suitable for both novice and experienced users.

The project does not require any external dependencies, databases, or backend services, as it is purely front-end based. The code is lightweight and easy to modify, allowing for potential future enhancements such as adding multimedia content, navigation menus, or animations. The SRS ensures that the project meets its objectives by clearly defining the required tools, system compatibility, and performance expectations.

CHAPTER 3

SYSTEM DESIGN

3. System Design

The RYTHM RISE project uses a simple system design with HTML for structure and CSS for styling. It includes a header with the website title, interactive buttons, and background styling. The design ensures a clean layout and smooth user interaction. It is also scalable, allowing for future enhancements.

3.1 Modules of System

System Components:

1. Music Program: A 12-week music intervention consisting of weekly sessions, workshops, and performances.

2. Youth Participants: 20-30 youth (ages 13-18) from diverse backgrounds, recruited through local schools, community centers, and social media.

3. Mentors/Instructors: 2-3 experienced music educators/instructors providing guidance, support, and feedback.

4. Community Partners: Local organizations, schools, and businesses providing resources, venues, and promotional support.

5. Evaluation Team: Researchers/evaluators collecting data, conducting surveys, and assessing program impact.

System Processes:

1. Recruitment: Youth participants and mentors/instructors are recruited and selected.

2. Music Program Delivery: Weekly music sessions, workshops, and performances are conducted.

3. Mentorship: Mentors/instructors provide guidance, support, and feedback to youth participants.

4. Community Engagement: Community partners provide resources, venues, and promotional support.

5. Evaluation: Data collection, surveys, and assessments are conducted to evaluate program impact.

6. Feedback and Improvement: Feedback from youth participants, mentors/instructors, and community partners is collected and used to improve the program.

System Stakeholders:

1. Youth Participants: Beneficiaries of the program, providing feedback and insights.

2. Mentors/Instructors: Delivering the music program, providing guidance and support.

3. Community Partners: Providing resources, venues, and promotional support.

4. Evaluation Team: Conducting research, collecting data, and assessing program impact.

5. Project Manager: Overseeing program implementation, coordination, and evaluation.

System Outputs:

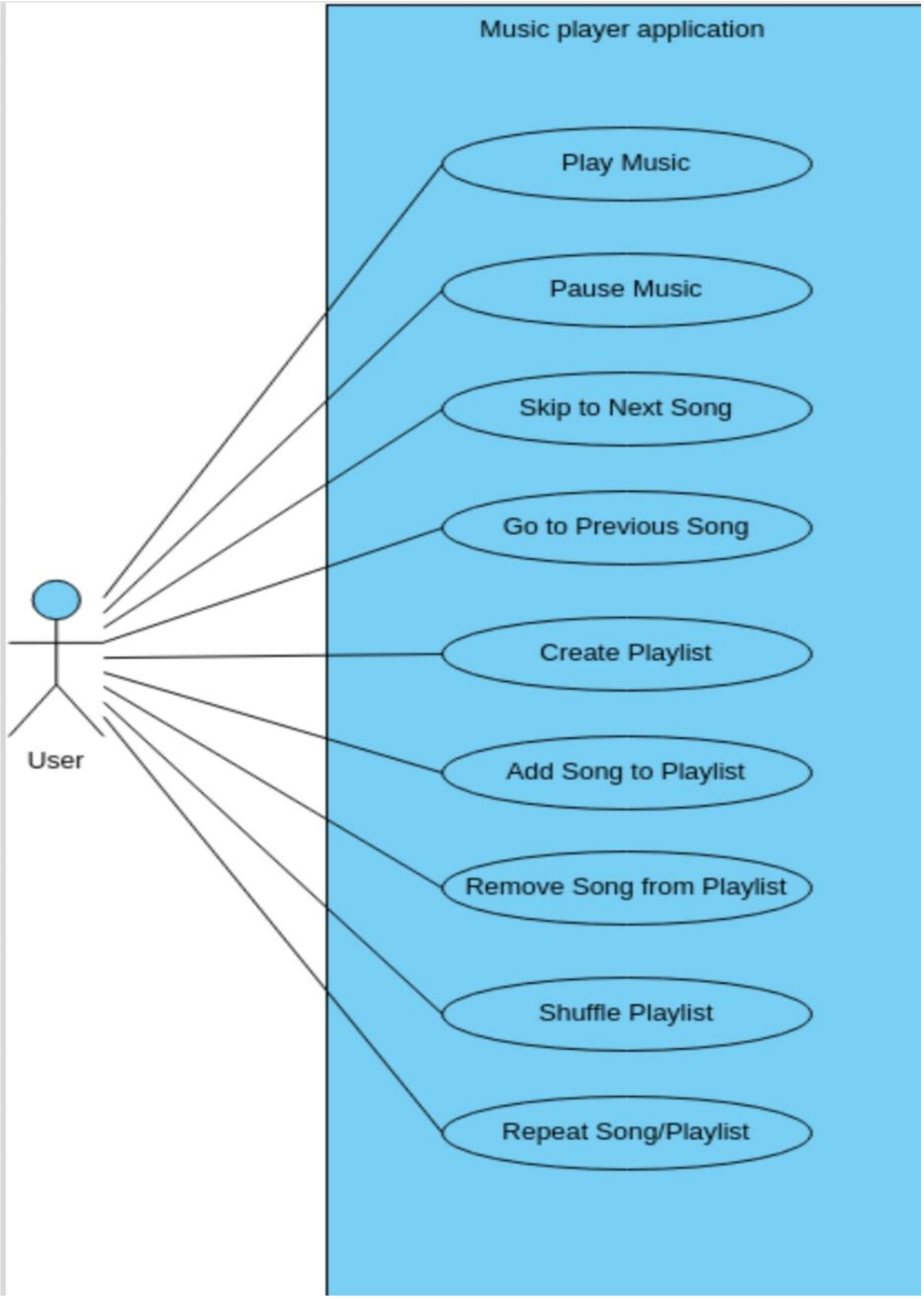
1. Improved Youth Development: Enhanced self-esteem, confidence, and life skills among youth participants.

2. Increased Community Engagement: Strengthened relationships between youth, community partners, and local organizations.

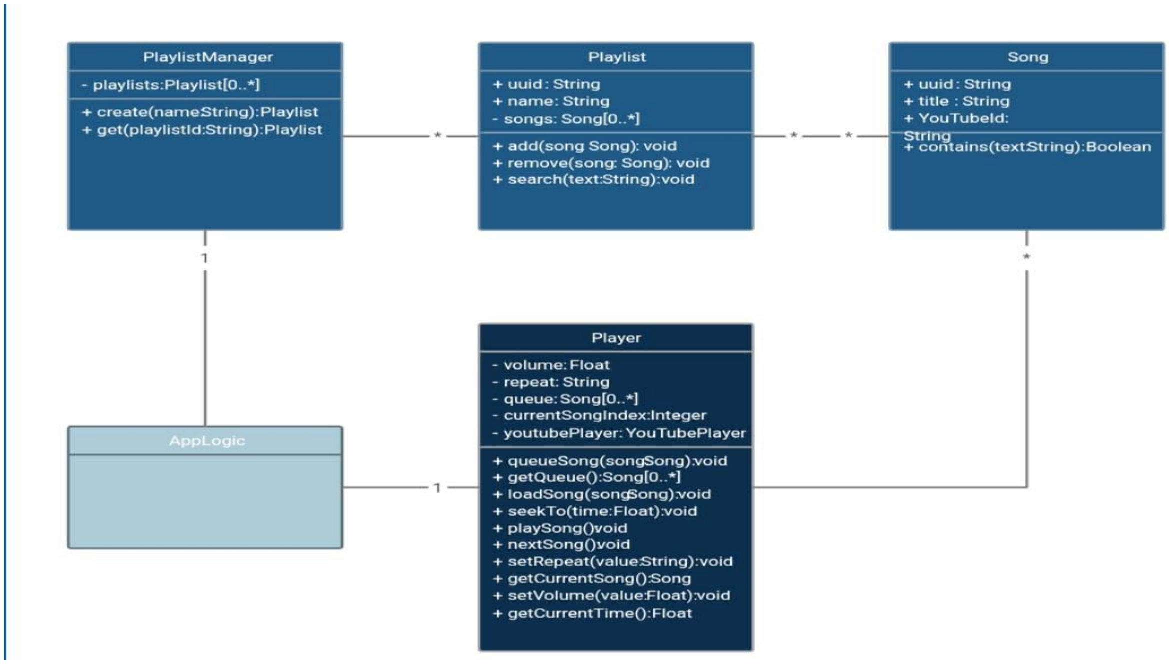
3. Music-Based Interventions: Development of effective music-based interventions for youth development.

4. Research and Evaluation: Contributions to the field of music therapy, youth development, and community engagement.

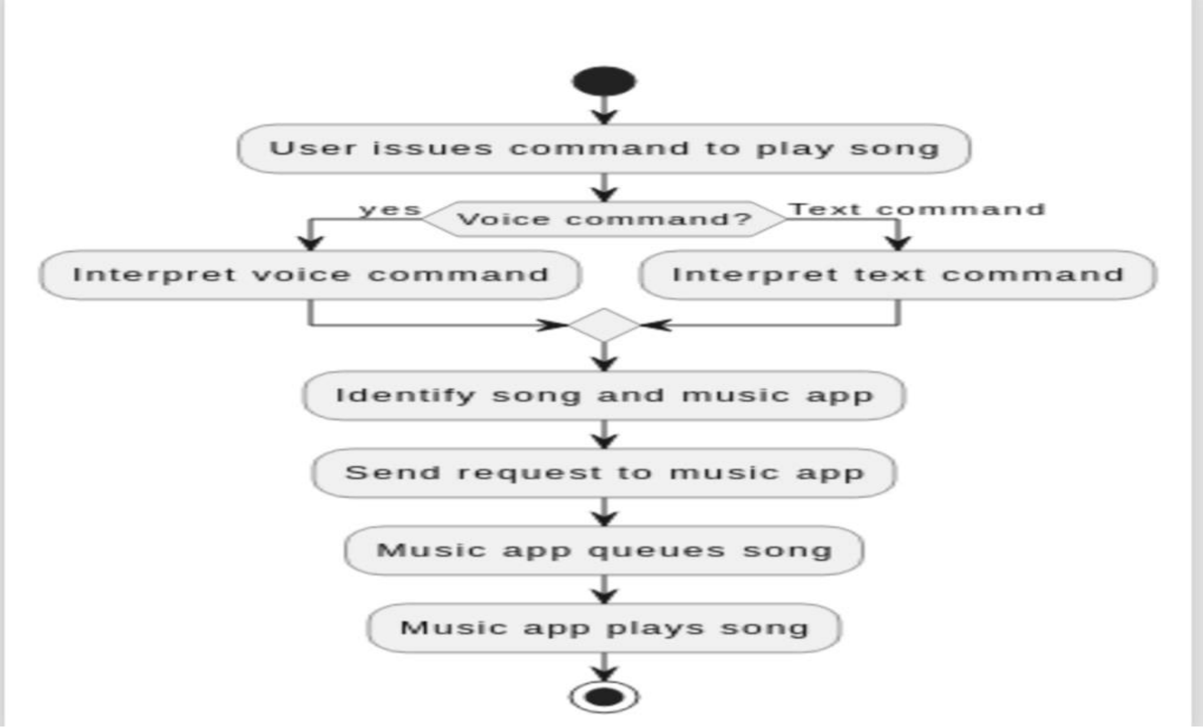
3.2 UML Diagrams



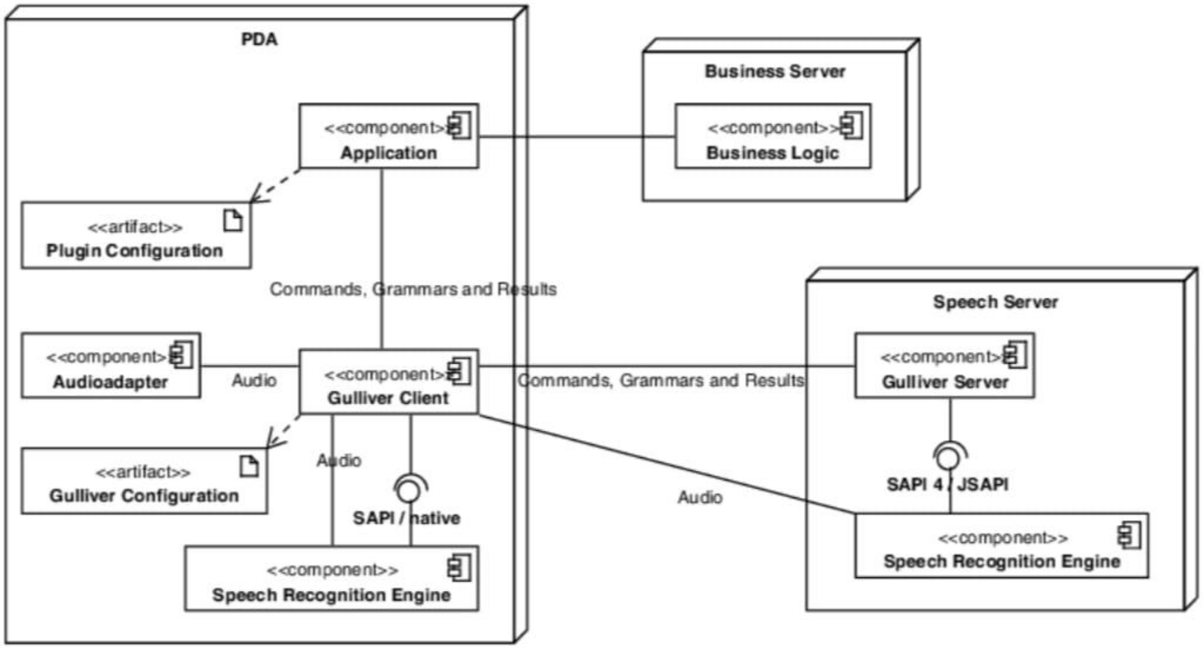
Fig(i): Use Case diagram


PlaylistManager
- playlists:Playlist[0..*]
+ create(nameString):Playlist
+ get(playlistId:String):Playlist
Playlist
+ uuid : String
+ name: String
- songs: Song[0..*]
+ add(song Song): void
+ remove(song: Song): void
+ search(text:String):void
Song
+ uuid : String
+ title : String
+ YouTubeId:
String
+ contains(textString):Boolean
*
*
*
1
*
Player
- volume: Float
- repeat: String
- queue: Song[0..*]
- currentSongIndex:Integer
- youtubePlayer: YouTubePlayer
+ queueSong(songSong):void
+ getQueue():Song[0..*]
+ loadSong(songSong):void
+ seekTo(time:Float):void
+ playSong()void
+ nextSong()void
+ setRepeat(valueString):void
+ getCurrentSong():Song
+ setVolume(value:Float):void
+ getCurrentTime():Float
AppLogic
1

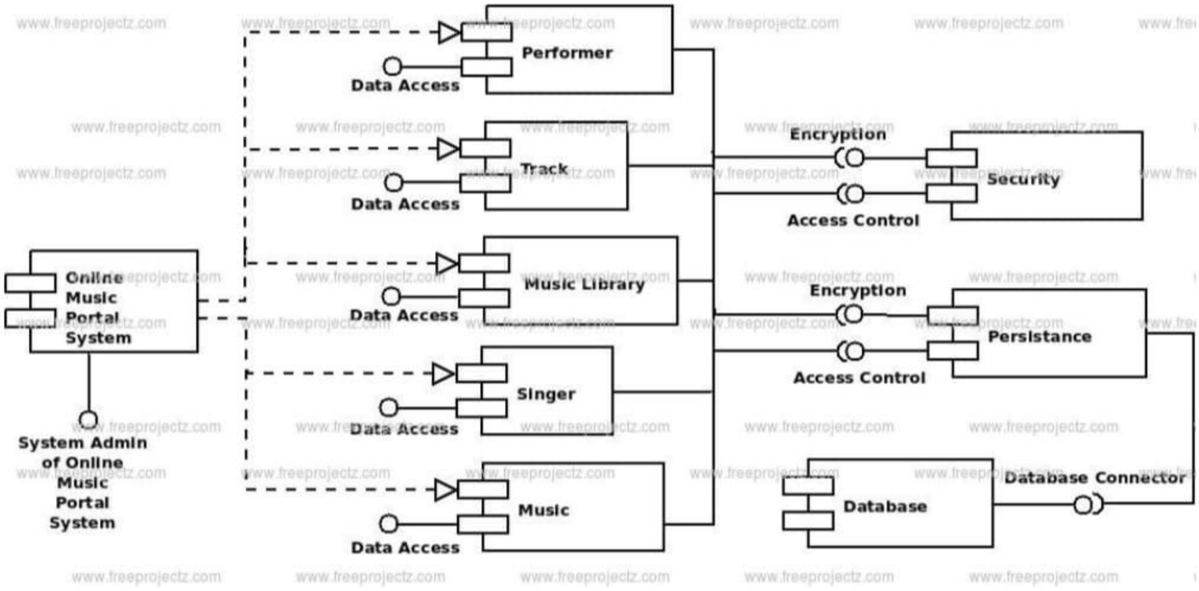
Fig(ii): Class diagram



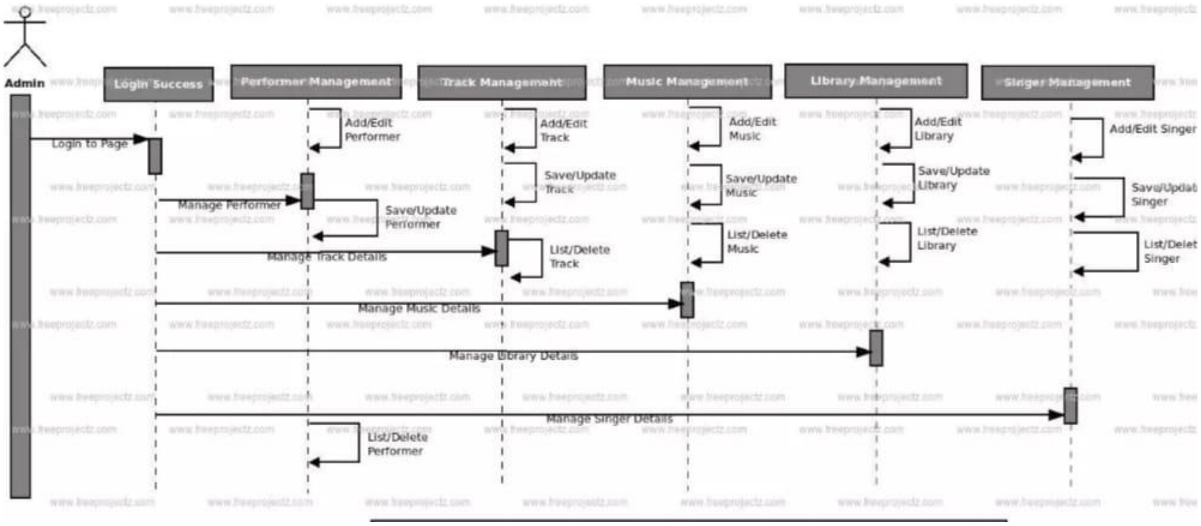
Fig(iii): Activity Diagram



Fig(iv): Communication Diagram



Fig(v): Component UML Diagram



Fig(vi): Sequence UML Diagram

CHAPTER 04

IMPLEMENTATION

4.Implementation

The RYTHM RISE project is implemented using HTML for the website structure and CSS for styling. The website includes a header, buttons, and background styling to create a clean and interactive layout. Hover effects on the buttons add basic interactivity, making the site more engaging. The code is lightweight and easy to modify, allowing for future improvements.

4.1 Sample Code

The following code represents the core structure of the RYTHM RISE music website. It uses HTML to define the content and layout, including the header, buttons, and basic page structure. The CSS styling enhances the visual appearance by adding colors, padding, and hover effects, making the website more interactive and user-friendly. The code is lightweight, easy to read, and can be expanded with additional features in the future.

CSS:

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <style>
    * {
      margin: 0;
padding: 0;
box-sizing: border-box;
      font-family: Arial, sans-serif;
    }
```

```
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-image: url('https://images.unsplash.com/photo-1614149162883-504ce4d13909?q=80&w=1974&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D');
    background-size: cover;
    background-position: center;
    color: rgba(255, 255, 255, 0.9);
}
.login-container {
    background: rgba(255, 255, 255, 0.2);
    padding: 25px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}
   input {
    width: 90%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fafafa;
```

```
        color: #333;
    }
. forgot-password {
        display: block;
        text-decoration: none;
        color: crimson;
        font-size: 14px;
        margin-bottom: 10px;
    }
    button {
        width: 100%;
        padding: 10px;
        background: crimson;
        color: white;
        border: none;
        cursor: pointer;
        border-radius: 5px;
       font-size: 16px;
        transition: background 0.3s ease;
    }
    button: hover {
        background-color: dark red;
    }
   .nav {
        position: absolute;
        top: 20px;
        right: 20px;
    }
    .nav li {
        display: inline;
        list-style: none;
    }
    .nav a {
        text-decoration: none;
        font-size: 1rem;
        font-weight: bold;
        padding: 10px 20px;
        color: rgb(252, 250, 250);
        border-radius: 5px;
        margin-left: 10px;
        transition: background 0.3s ease;
    }
.nav a: hover {
        background: rgba(26, 20, 21, 0.444);
    }
   . message {
        font-size: 14px;
        margin-top: 10px;
    }
     . register-link {
        margin-top: 10px;
        font-size: 14px;
        color: #f9f9f9;
    }
    . register-link a {
      color: crimson;
      text-decoration: none;
      font-weight: bold;
    }
    . register-link a: hover {
      text-decoration: underline;
    }
  </style>
</head>
<body>

<! -- Login Form -->
<div class="login-container">
  <h2>Login</h2>
  <form id="loginForm">
    <input type="text" id="username" placeholder="Username" required />
  <input type="password" id="password" placeholder="Password" required />
    <button type="submit">Login</button>
    <div class="message" id="loginMessage"></div>
   <div class="register-link">
      Don't have an account? <a href="registration.html">Register</a>
    </div>
  </form>
</div>
<script>
  // Event Listener for Login Form
  document.getElementById('loginForm').addEventListener('submit', async function(event) {
  event.preventDefault();
 const username = document.getElementById('username').value.trim();
  const password = document.getElementById('password').value.trim();
  const response = await fetch("http://localhost:5000/login", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ username, password })
  });
const result = await response.json();
if (response.ok) {
   localStorage.setItem("token", result.token); // Store JWT token for authentication
    localStorage.setItem("username", result.username);
    window.location.href = "home.html"; // Redirect after login
  } else {
    alert(result.message);
  }
});
</script>
</body>
</html>
```

Register Page:

```
<div class="register-container">
  <h2>Register</h2>
  <form id="registerForm">
    <input type="text" id="username" placeholder="Username" required />
    <input type="email" id="email" placeholder="Email" required />
    <input type="password" id="password" placeholder="Password" required />
    <button type="submit">Register</button>
    <div class="message" id="registerMessage"></div>
    <div class="login-link">
Already have an account? <a href="login.html">Login</a>
    </div>
  </form>
</div>
<script>
  document.getElementById('registerForm').addEventListener('submit', async function(event) {
    event.preventDefault();
    // Get user input
    const username = document.getElementById('username').value.trim();
    const email = document.getElementById('email').value.trim();
    const password = document.getElementById('password').value.trim();
    // Send data to backend
    const response = await fetch('http://localhost:5000/register', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ username, email, password })
    });
    const result = await response.json();
if (response.ok) {
      document.getElementById('registerMessage').style.color = 'green';
      document.getElementById('registerMessage').innerText = result.message;
```

```
            // Redirect to login after 2 seconds
            setTimeout(() => {
                window.location.href = 'login.html';
            }, 2000);
        } else {
            document.getElementById('registerMessage').style.color = 'red';
            document.getElementById('registerMessage').innerText = result.message;
        }
    });
</script>

</body>
</html>
```

Home page

```
<body>
    <script src="script.js"></script>
    <div class="container">
        <ul class="nav">
            <li><a href="#" id="logoutLink">LOGOUT</a></li>
            <li><a href="contact.html" style="font-size:100%;">CONTACT US</a></li>
            <li><a href="about.html" style="font-size:100%;">ABOUT US</a></li>
            <li><a href="artist.html"style="font-size:100%;">ARTISTS</a></li>
            <li><a href="music.html" style="font-size:100%;">MUSIC</a></li>
            <li><a href="home.html" style="font-size:100%;">HOME</a></li>
 </ul>
 </div>
```

```
<div class="content">
        <div class="left-col">
            <h1 style="color: #141413;">RYTHM<br>RISE</h1>
        </div>
</div>
<audio id="mysound">
        <source src="media/music.mp3" type="audio/mp3">
    </audio>
<script>
        let mysound = document.getElementById("mysound");
        let icon = document.getElementById("icon");

// Creating a function that will change
        // pause to play and vice-versa
        icon.onclick = function () {
            if (mysound.paused) {

                // If paused then play the
                // music and change the image
                mysound.play();
                icon.src =
“https://media.geeksforgeeks.org/wp-content/uploads/20210402235545/Pause.png”;
            } else {

                // If playing then pause the
                // music and change the image
                mysound.pause();
            icon.src =
“https://media.geeksforgeeks.org/wp-content/uploads/20210402235520/play.png”;
          }
        }
    </script>

</body>

</html>
```

Music CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Music - RYTHM RISE</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: Arial, sans-serif;
        }

        body {
                background-image: url('https://images.unsplash.com/photo-1614149162883-
504ce4d13909?q=80&w=1974&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D');
```

```
          background-size: cover;
          background-position: center;
          color: rgb(14, 12, 12);
        }
      /* Scroll Navigation Bar Styles */
.scroll-navbar {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    background-color: rgba(0, 0, 0, 0.8);
    padding: 10px;
    display: flex;
    justify-content: center;
    gap: 15px;
    z-index: 1000;
}

.scroll-navbar button {
    background-color: #444;
    color: white;
    border: none;
    padding: 8px 12px;
    cursor: pointer;
    border-radius: 4px;
    transition: background 0.3s;
}
```

```
.scroll-navbar button:hover {
  background-color: #666;
}

      .container {
        width: 80%;
        margin: auto;
        padding: 50px 0;
        text-align: center;
      }

      .music-heading {
        font-size: 48px;
        font-weight: bold;
          color: rgb(235, 224, 226);
        margin-bottom: 20px;
      }

      .category {
        margin-top: 40px;
        text-align: left;
      }

      .category h2 {
        color: rgb(253, 245, 245);
        font-size: 28px;
```

```
    border-bottom: 2px solid rgb(16, 15, 15);
    padding-bottom: 5px;
    margin-bottom: 20px;
}

.music-list {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    gap: 20px;
}

.music-card {
    background: rgba(252, 244, 244, 0.438);
    padding: 15px;
    border-radius: 10px;
    width: 250px;
    text-align: center;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
}

.music-card img {
    width: 100%;
    border-radius: 8px;
}

.music-card h3 {
    margin: 10px 0;
    font-size: 18px;
    color: rgb(28, 21, 21);
}

.music-card audio {
    width: 100%;
    margin-top: 10px;
}

.footer {
    text-align: center;
    padding: 20px;
    font-size: 14px;
    margin-top: 40px;
}
.nav {
    position: absolute;
    top: 20px;
    right: 20px;
}

.nav li {
    display: inline;
    list-style: none;
}
```

```
    .nav a {
      text-decoration: none;
      font-size: 1rem;
      font-weight: bold;
      padding: 10px 20px;
      color: rgb(194, 237, 5);
      border-radius: 5px;
      margin-left: 10px;
    }
    .nav a:hover {
      background: rgba(25, 17, 19, 0.434);}
      #queue {
      margin-top: 20px;
      padding: 10px;
      border: 2px solid #000;
      min-height: 50px;
    }
    .queue-item {
      display: inline-block;
      padding: 10px;
      margin: 5px;
      background-color: #f0f0f0;
      border-radius: 5px;
    }
    /* Music Navbar Styling */
    .music-container {
  display: flex;
```

```
    flex-direction: column;
    align-items: center; /* Centers links horizontally */
    gap: 10px; /* Adds spacing between links */
    margin-top: 20px;
 /* Black background for the container */
    padding: 20px;
    border-radius: 10px; /* Optional: rounded corners for a sleek look */
}

.music-container a {
    width: 200px; /* Sets a fixed width for all links */
    text-align: center;
    padding: 10px;
    font-size: 16px;
    text-decoration: none;
    background-color: transparent; /* No background for links */
    color: white; /* White text color */
    border: 2px solid white; /* White border for a button-like effect */
    border-radius: 5px;
    transition: background-color 0.3s ease, color 0.3s ease;
}

.music-container a:hover {
    background-color: white; /* White background on hover */
    color: black; /* Black text on hover */
}
```

```
/* Smooth Scroll */
html {
    scroll-behavior: smooth;
}


    </style>
</head>
```

Music Page:

```
<body>
 <script src="script.js"></script>
    <ul class="nav">
        <li><a href="home.html">HOME</a></li>
        <li><a href="music.html">MUSIC</a></li>
        <li><a href="artist.html">ARTISTS</a></li>
        <li><a href="about.html">ABOUT US</a></li>
        <li><a href="contact.html">CONTACT US</a></li>
        <li><a href="#" id="logoutLink">LOGOUT</a></li>
  </ul>
    <div class="container">
        <h1 class="music-heading">Explore Music - RYTHM RISE</h1>
           <p style="color: white;">Listen to your favorite tracks, discover new music, and enjoy unlimited streaming.</p>
           <div class="music-container">
            <a href="sad-songs.html">Sad Songs</a>
            <a href="romantic-songs.html">Romantic Songs</a>
            <a href="love-songs.html">Love Songs</a>
```

```
      <a href="hits-20s.html">20's Hits</a>
      <a href="telugu-melodies.html">Telugu Melodies</a>
    </div>

  <div class="footer">
    <p style="color: white;">&copy; 2025 RYTHM RISE | All Rights Reserved</p>
  </div>
</div>
<script>
    let playlist = [], queue = [], repeatMode = false, shuffleMode = false;
    function addSongToPlaylist() {
      let file = document.getElementById("songInput").files[0];
      if (file) {
        let url = URL.createObjectURL(file);
        playlist.push({ name: file.name, url: url });
        updatePlaylistUI();
      }
    }
    function updatePlaylistUI() {
      let playlistUI = document.getElementById("playlist");
      playlistUI.innerHTML = "";
      playlist.forEach((song, index) => {
        let li = document.createElement("li");
        li.textContent = song.name;
        li.onclick = () => playSong(index);
        playlistUI.appendChild(li);
      });
```

```
 }
 function playSong(index) {
     audioPlayer.src = playlist[index].url;
     audioPlayer.play();
 }


 function toggleShuffle() {
     shuffleMode = !shuffleMode;
     alert(shuffleMode ? "Shuffle Mode ON" : "Shuffle Mode OFF");
     if (shuffleMode) {
         playlist.sort(() => Math.random() - 0.5);
         updatePlaylistUI();
     }
 }
function scrollToSection(sectionId) {
     document.getElementById(sectionId).scrollIntoView({ behavior: 'smooth' });
 }
 function toggleRepeat() {
     repeatMode = !repeatMode;
     alert(repeatMode ? "Repeat Mode ON" : "Repeat Mode OFF");
     audioPlayer.loop = repeatMode;
 }
  function downloadSong() {
     let link = document.createElement("a");
     link.href = audioPlayer.src;
     link.download = "song.mp3";
     document.body.appendChild(link);
```

```
            link.click();
            document.body.removeChild(link);
        }
 </script>
</body>
</html>
```

Source Code Telugu melodies page:

```
<body>
   <div id="nav">
        <button id="myPlaylistBtn" onclick="goToPlaylist()">🎵 My Playlist</button>
    </div>
 <!-- Telugu Melodies Section -->
 <div class="category">
    <h2>🎶 Telugu Melodies</h2>
    <div class="music-list">
        <div class="music-card">
            <img src="https://qqcdnpictest.mxplay.com/pic/9053ad83c5b2e34bdd3ceb1079ec6c9e/te/16x9/640x360/test_pic1577694194918.jpg" alt="Song Cover">
            <h3>Gagana Veedhilo</h3>
            <p> Music : Mickey J Meyer </p>
            <audio controls>
                <source src="playlist/[iSongs.info] 02 - Gagana Veedhilo.mp3" type="audio/mp3">
                Your browser does not support the audio element.
            </audio>
            <div>
              <span class="controls" onclick="addToPlaylist('Gagana Veedhilo', 'playlist/[iSongs.info] 02 - Gagana Veedhilo.mp3')">➕</span>
              <a href="playlist/[iSongs.info] 02 - Gagana Veedhilo.mp3" download="Gagana_Veedhilo.mp3">⬇ </a>


            </div>


        </div>


        <div class="music-card">
              <img src="https://files.prokerala.com/movies/pics/800/tamanna-ram-in-endukante-premanta-9038.jpg" alt="Song Cover">
            <h3>Nee Choopule Song</h3>
            <p>Music : G V Prakash Kumar</p>
            <audio controls>
                <source src="playlist/[iSongs.info] 02 - Nee Choopule.mp3" type="audio/mp3">
                Your browser does not support the audio element.
            </audio>
            <div>
                <span class="controls" onclick="addToPlaylist('Nee Choopule Song', 'playlist/[iSongs.info] 02 - Nee Choopule.mp3')">➕</span>
                <a href="playlist/[iSongs.info] 02 - Nee Choopule.mp3" download="Nee_Choopule.mp3">⬇ </a>
             </div>
          </div>
    <div class="music-card">
<img src="https://sund-images.sunnxt.com/101537/1920x1080_Majili_101537_a8924cb2-d834-4ba4-8d11-0d79c2e9d849.jpg" alt="Song Cover">
             <h3>Priyathama Priyathama</h3>
            <p>Music: Gopi Sunder </p>
            <audio controls>
              <source src="playlist/[iSongs.info] 02 - Priyathama Priyathama.mp3" type="audio/mp3">
                Your browser does not support the audio element.
            </audio>
            <div>
        <span class="controls" onclick="addToPlaylist('Priyathama Priyathama', 'playlist/[iSongs.info] 02 - Priyathama Priyathama.mp3')">➕</span>
            <a href="playlist/[iSongs.info] 02 - Priyathama Priyathama.mp3" download="Priyathama_Priyathama.mp3">⬇ </a>
             </div>
        </div>
        <div class="music-card">
             <img src="https://tse2.mm.bing.net/th?id=OIP.YbU1mm63G2UuI9FDgOhFhgHaEK&pid=Api&P=0&h=180" alt="Song Cover">
            <h3>Srimathi Garu</h3>
            <p>Music : G V Prakash Kumar</p>
            <audio controls>
                  <source src="playlist/[iSongs.info] 01 - Srimathi Garu.mp3.crdownload" type="audio/mp3">
                Your browser does not support the audio element.
            </audio>
            <div>
                <span class="controls" onclick="addToPlaylist('Srimathi Garu', 'playlist/[iSongs.info] 01 - Srimathi Garu.mp3')">➕</span>
         <a href="playlist/[iSongs.info] 01 - Srimathi Garu.mp3" download="Srimathi_Garu.mp3">⬇ </a>
```

```
          </div>


        </div>
        <div class="music-card">
          <img src="https://tse4.mm.bing.net/th?id=OIP.gEW016f_iF1uRKhjnNy9QAHaEj&pid=Api&P=0&h=180" alt="Song Cover">
          <h3>Nee Kallalona</h3>
          <p>Music : G V Prakash Kumar</p>
          <audio controls>
            <source src="playlist/[iSongs.info] 03 - Nee Kallalona.mp3" type="audio/mp3">
            Your browser does not support the audio element.
          </audio>
          <div>

            <span class="controls" onclick="addToPlaylist('Nee Kallalona', 'playlist/[iSongs.info] 03 - Nee Kallalona.mp3')">➕</span>
            <a href="playlist/[iSongs.info] 03 - Nee Kallalona.mp3" download="Nee_Kallalona.mp3">⬇ </a>

          </div>


        </div>
   </div>
</div>
</div>
<!--
<div id="playlist-container">
```

```
    <h2>📜 Playlist</h2>
    <ul id="playlist"></ul>
</div>-->
 <!-- Queue Display -->
 <!-- <div>
    <h2>📝 Queue</h2>
    <ul id="queue-list"></ul>
</div>
<h3 id="current-song-title">Now Playing: None</h3>
    <audio id="audio-player" controls></audio>-->
    <h2>Music Queue <span onclick="toggleRepeat()">🔁</span> <span onclick="toggleShuffle()">🔀</span></h2>

    <p id="current-song-title">Now Playing: None</p>

    <audio id="audio-player" controls></audio>
    <div id="queue-container">
        <h3>Queue List</h3>
        <div id="queue-list"></div>
    </div>
<div class="footer">
    <p style="color: white;">&copy; 2025 RYTHM RISE | All Rights Reserved</p>
</div>>

    <script>
        function goToPlaylist() {
            window.location.href = "playlist.html"; // Redirects to the playlist page
        }
      let songQueue = [
            { title: "Gagana Veedhilo", src: "playlist/[iSongs.info] 02 - Gagana Veedhilo.mp3" },
            { title: "Nee Choopule Song", src: "playlist/[iSongs.info] 02 - Nee Choopule.mp3" },
            { title: "Priyathama Priyathama", src: "playlist/[iSongs.info] 02 - Priyathama Priyathama.mp3" },
            { title: "Srimathi Garu", src: "playlist/[iSongs.info] 01 - Srimathi Garu.mp3.crdownload" },
            { title: "Nee Kallalona", src: "playlist/[iSongs.info] 03 - Nee Kallalona.mp3" }
        ];
        let currentIndex = null;
        let isRepeat = false;
        let isShuffle = false;
        let playedSongs = [];
        let audioPlayer = document.getElementById("audio-player");
        function loadSong(index) {
            if (index >= 0 && index < songQueue.length) {
              audioPlayer.src = songQueue[index].src;
              document.getElementById("current-song-title").innerText = "Now Playing: " + songQueue[index].title;
              audioPlayer.play();
            }
        }
        function toggleRepeat() {
            isRepeat = !isRepeat;
            alert("Repeat Mode: " + (isRepeat ? "ON - Playing only current song" : "OFF"));
        }
        function toggleShuffle() {
            isShuffle = !isShuffle;
        playedSongs = []; // Reset played songs list when shuffle starts
        alert("Shuffle Mode: " + (isShuffle ? "ON - Playing songs in random order" : "OFF"));
}
function playNextSong() {
    if (isRepeat) {
        loadSong(currentIndex); // Play the same song again
    } else if (isShuffle) {
        if (playedSongs.length === songQueue.length) {
            playedSongs = []; // Reset shuffle queue if all songs played
        }

        let nextIndex;
        do {
            nextIndex = Math.floor(Math.random() * songQueue.length);
        } while (playedSongs.includes(nextIndex)); // Ensure a song is not repeated in the current shuffle session

        playedSongs.push(nextIndex);
        currentIndex = nextIndex;
        loadSong(currentIndex);
    } else {
        currentIndex = (currentIndex + 1) % songQueue.length;
        loadSong(currentIndex);
    }
}
audioPlayer.addEventListener("ended", playNextSong);
function updateQueueDisplay() {
    let queueList = document.getElementById("queue-list");
    queueList.innerHTML = "";

    songQueue.forEach((song, index) => {
        let songItem = document.createElement("p");
        songItem.innerHTML = `🎵 ${song.title}`;
        songItem.style.cursor = "pointer";
        songItem.addEventListener("click", () => {
            currentIndex = index;
            loadSong(index);
        });
        queueList.appendChild(songItem);
    });
}
updateQueueDisplay();
loadSong(currentIndex);

function addToPlaylist(title, src) {
    let playlist = JSON.parse(localStorage.getItem("playlist")) || [];

    if (!playlist.some(song => song.src === src)) {
        playlist.push({ title, src });
        localStorage.setItem("playlist", JSON.stringify(playlist));
        alert(`"${title}" added to your playlist!`);
    } else {
        alert(`"${title}" is already in the playlist.`);
    }
```

```
        }
        document.addEventListener("keydown", function (event) {
        if (event.key === "Enter" && !audioPlayer.paused) {
            audioPlayer.pause();
            if (currentPlaying) {
                currentPlaying.classList.remove("active");
            }
        }
    });
    </script>

</body>
</html>
```

SOURCE CODE:- Artist

```
<head>
    <style>
        artist-container {
            display: flex;
            flex-wrap: wrap;
            justify-content: center;
          margin-top: 20px;
        }
        .artist {
            background: rgba(255, 255, 255, 0.458);
        padding: 20px;

        margin: 15px;

        border-radius: 10px;

        width: 250px;

        text-align: center;

    }

    .artist h2 {

        color: rgb(31, 10, 14);

        font-size: 1.5rem;

        margin-bottom: 10px;

    }

    .artist ul {

        list-style: none;

        padding: 0;

    }

    .artist ul li {

        font-size: 1rem;

        color: black;
   }

    li:hover, li.active {

        background: rgba(255, 255, 255, 0.3);

    }

</style>
```

```
</head>

<body>

    <audio id="audioPlayer" controls style="display:none;"></audio>

    <script src="script.js"></script>

    <script>

        let currentPlaying = null;

        function togglePlay(element, songPath) {

            let audioPlayer = document.getElementById("audioPlayer");

            if (currentPlaying === element) {

                if (!audioPlayer.paused) {

                    audioPlayer.pause();

                    element.classList.remove("active");

                } else {

                    audioPlayer.play();

                    element.classList.add("active");

                }
            } else {

                if (currentPlaying) {
                    currentPlaying.classList.remove("active");

     }
                currentPlaying = element;

                audioPlayer.src = songPath;

                audioPlayer.play();
                element.classList.add("active");
            }
        }
    </script>
    <div class="container">
        <ul class="nav">
            <li><a href="#" id="logoutLink">LOGOUT</a></li>
            <li><a href="contact.html">CONTACT US</a></li>
            <li><a href="about.html">ABOUT US</a></li>
            <li><a href="artist.html">ARTISTS</a></li>
            <li><a href="music.html">MUSIC</a></li>
            <li><a href="login.html">LOGIN</a></li>
            <li><a href="home.html">HOME</a></li>
        </ul>
    </div>
    <div class="content">
        <h1>Featured Artists</
h1>
        <div class="artist-container">
            <div class="artist">
             <imgsrc="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ2wQi_hOm9J-
eAljatY8dkXfW7o_q7YPEWC-_hGnBjcB2usu56RXAK15R9R4GbJZojMSw&usqp=CAU"
alt="Song Cover">
```

```
            <h2>Anirudh</h2>

            <ul>

              <li onclick="togglePlay(this,'playlist/[iSongs.info] 02 - Gaali Vaaluga.mp3')">Gaali
Vaaluga</li>

              <li onclick="togglePlay(this,'playlist/[iSongs.info] 01 - The Karma
Theme.mp3')">The karma Theme</li>

              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 05 - Luckkanna Mate
Nillu.mp3')">Luckkanna Mate Nillu</li>

              <li onclick="togglePlay(this, 'playlist/Thangamagan - Enna Solla Lyric _ Anirudh
Ravichander _ Dhanush.m4a')">Enna Solla</li>

              <li onclick="togglePlay(this, 'playlist/Dheema - Masstamilan.mp3')">Dheema
Dheema</li>

              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 02 -
Kollagotte.mp3')">Kollagottey</li>

              <li onclick="togglePlay(this, 'playlist/Hukum Tiger -
Masstamilan.mp3')">Hukum</li>

              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 01 - Master
Coming.mp3')">Master Coming</li>

            </ul>

          </div>

          <div class="artist">

            <imgsrc="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRDZB-
uYKghbeHZLa5ZpRExBRML-PSvCzzg-
ES27tv6N0ktQC2TDtkpOlyVMqIYxkgm03w&usqp=CAU" alt="Song Cover">

            <h2>Shreya Ghoshal</h2>
            <ul>
              <lionclick="togglePlay(this, 'playlist/[iSongs.info] 02 - Sooseki.mp3')">Sooseki</li>
              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 02 - Digu Digu Digu Naaga.mp3')">Digu Digu Digu Naaga</li>
              <li onclick="togglePlay(this, 'playlist/Tujh Mein Rab Dikhta Hai _ Female Version _ Rab Ne Bana Di Jodi, SRK, Anushka Sharma, Shreya Ghoshal.m4a')">Tujhme Rab Dikhta Hai</li>
              <li onclick="togglePlay(this, 'playlist/Chaudhavi Shab _ Video Song _ Sanjay Leela Bhansali _ Shreya Ghoshal _ Heeramandi _ Bhansali Music.m4a')">Chaudhavi Shab</li>
              <li onclick="togglePlay(this, 'playlist/Chikni Chameli Best Video - Agneepath _ Katrina, Hrithik _ Shreya _ Ajay-Atul.m4a')">Chikni Chameli</li>
              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 01 - Itu Itu Ani Chitikelu Evvarivo.mp3')">Itu Itu AniChitikelu Evvarivo</li>
            </ul>
        </div>
        <div class="artist">
            <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSNmjkdULf4Dy6MXJKmOvfwVy3kgN4t2bzdRg&s" alt="Song Cover">
            <h2>Karthik</h2>
            <ul>
              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 02 - Ram Sita Ram.mp3')">Ram Sita Ram</li>
               <li onclick="togglePlay(this, 'playlist/[iSongs.info] 01 - Niluvadhamu Ninu Epudaina.mp3')">Niluvaddham</li>
              <li onclick="togglePlay(this, 'playlist/Beautiful Girl Full Video Song - Life is Beautiful Movie Video Songs.m4a')">Beautiful Girl</li>

              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 02 - Chilipiga.mp3')">Chilipiga</li>

              <li onclick="togglePlay(this, 'playlist/[iSongs.info] 04 - Priya Mithunam.mp3')">Priya Mithunam</li>

              <li onclick="togglePlay(this, 'playlist/Usurey poguthu Song lyrics in தமிழ் _AR Rahman _Ravanan _Tamil Entertain_.m4a')">Usurey Pogudhey</li>

              <li onclick="togglePlay(this, 'playlist/NaaNaa Hyraanaa - Lyrical  _ Game Changer _ Ram Charan _ Kiara _ Shreya Ghoshal_ Karthik _ Thaman.m4a')">NaaNaa Hyraanaa</li>

            </ul>

        </div>

        <div class="artist">

            <img src="https://static.toiimg.com/thumb/msid-96585261,imgsize-57974,width-400,resizemode-4/96585261.jpg" alt="Arijit Singh" style="width: 210px; height: 250px;">

            <h2>Arjit Singh</h2>

            <ul>

              <li onclick="togglePlay(this, 'playlist/kanulanu thaake (8d audio).m4a')">Kanulanu Thaake</li>

              <li onclick="togglePlay(this, 'playlist/Heeriye (Official Video) Jasleen Royal ft Arijit Singh_ Dulquer Salmaan_ Aditya Sharma _Taani Tanvir.m4a')">Heeriye</li>

              <li onclick="togglePlay(this, 'playlist/Chaleya (Lyrics) - Jawan _ Shah Rukh Khan _ Nayanthara _ Atlee, Anirudh _ Arijit Singh _ Shilpa Rao.m4a')">Chaleya</li>

             <li onclick="togglePlay(this, 'playlist/[iSongs.info] 04 - Anuvanuvuu.mp3')">Anuvanuvuu</li>
```

```
                    <li onclick="togglePlay(this, 'playlist/Tere Pyaar Mein (Full Video) Tu Jhoothi Main Makkaar_ Ranbir,Shraddha_ Pritam_Arijit,Nikhita,Amitabh.m4a')">Tere Pyaar Mein</li>
                    <li onclick="togglePlay(this, 'playlist/ANIMAL- Satranga (Lyrics) _ Arijit _ Ranbir Kapoor _ Rashmika.m4a')">Satranga</li>
                    <li onclick="togglePlay(this, 'playlist/What Jhumka _ (Lyrics) - Rocky Aur Rani Kii Prem Kahani _ Ranveer _ Alia _ Arjit Singh _ Pritam _.m4a')">What jhumka?</li>
                </ul>
            </div>
        </div>
    </div>
</body>
</html>
```

SOURCE CODE:-About

```
<head>
<style>
        .about-text {
            font-size: 18px;
            line-height: 1.6;
            max-width: 800px;
            margin: auto;
            background: rgba(0, 0, 0, 0.6);
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
        }
        .footer {
            text-align: center;
            padding: 20px;
            font-size: 14px;
            margin-top: 40px;
        }
    </style>
</head>
<body>
    <script src="script.js"></script>
    <div class="container">
        <ul class="nav">
            <li><a href="#" id="logoutLink">LOGOUT</a></li>
            <li><a href="contact.html" style="font-size:100%;">CONTACT US</a></li>
            <li><a href="about.html" style="font-size:100%;">ABOUT US</a></li>
            <li><a href="artist.html"style="font-size:100%;">ARTISTS</a></li>
            <li><a href="music.html" style="font-size:100%;">MUSIC</a></li>
            <li><a href="login.html" style="font-size:100%;">LOGIN</a></li>
            <li><a href="home.html">HOME</a></li>
        </ul>
    </div>
    <div class="container">
        <h1 class="about-heading">About Us - RYTHM RISE</h1>
        <p class="about-text">
            Welcome to <b>RYTHM RISE</b>, the ultimate destination for music lovers, artists, and creators!
            Our platform is designed to connect people through the power of music, providing a seamless
            and immersive experience for discovering, sharing, and enjoying songs from around the world.
            <br><br>
            ♪ <b>Discover New Music:</b> Explore a vast collection of tracks from emerging and established artists.<br>
            🎧 <b>Stream Anytime, Anywhere:</b> Enjoy high-quality audio streaming without interruptions.<br>
            🚀 <b>For Artists:</b> Upload your music, reach a global audience, and grow your fanbase.<br>
            ❤ <b>Create Your Playlists:</b> Customize your music journey with playlists for every mood.<br>
            🎙 <b>Live Performances:</b> Tune in to exclusive live sessions from talented musicians.<br>
            <br>
            At <b>RYTHM RISE</b>, we believe in the power of music to inspire, uplift, and bring people together.
            Whether you’re a listener or an artist, our mission is to provide you with an engaging and
              enjoyable music experience like never before.
            <br><br>
            Join us on this musical journey and let the rhythm rise!
        </p>
    </div>

    <div class="footer">
        <p>&copy; 2025 RYTHM RISE | All Rights Reserved</p>
    </div>
</body>
</html>
```

SOURCE CODE:-Contact

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Us</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-image:url('https://images.unsplash.com/photo-1614149162883-
504ce4d13909?q=80&w=1974&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D');
            background-size: cover;
        }
        .container {
            background: rgba(138, 131, 131, 0.445);
            padding: 50px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 300px;
        }
        h2 {
            text-align: center;
        }
        .form-group {
            margin-bottom: 15px;
        }
        label {
  display: block;
            margin-bottom: 5px;

  }
```

```
input, textarea {
   width: 100%;
   padding: 8px;
   border: 1px solid #ccc;
   border-radius: 5px;
}
button {
   width: 100%;
   padding: 10px;
   background: #28a745;
   color: white;
   border: none;
   border-radius: 5px;
   cursor: pointer;
}
button:hover {
   background: #218838;
}
.contact-info {
   text-align: center;
   margin-bottom: 20px;
}
.nav {
   position: absolute;
   top: 20px;
   right: 20px;
}
.nav li {
   display: inline;
   list-style: none;
}
```

```
        .nav a {
            text-decoration: none
           font-size: 1rem;
            font-weight: bold;
            padding: 10px 20px;
            color: rgba(210, 241, 10, 0.564);
            border-radius: 5px;
            margin-left: 10px;
            transition: background 0.3s ease;
        }
        .nav a:hover {
            background: rgba(82, 30, 39, 0.444);
        }
    </style>
</head>
<body>
    <script src="script.js"></script>
    <ul class="nav">
        <li><a href="home.html">HOME</a></li>
        <li><a href="music.html">MUSIC</a></li>
        <li><a href="artist.html">ARTISTS</a></li>
        <li><a href="about.html">ABOUT US</a></li>
        <li><a href="contact.html">CONTACT US</a></li>
        <li><a href="#" id="logoutLink">LOGOUT</a></li>
    </ul>
    <div class="container">
        <h2>Contact Us</h2>
        <div class="contact-info">
            <p><strong>Address:</strong> Vignan University</p>
            <p><strong>Phone:</strong> 7989644974</p>
            <p><strong>Email:</strong> 231fa04328@gmail.com</p>
        </div>
        <form onsubmit="sendMessage()" >
            <div class="form-group">
              <label for="fullname">Full Name</label>
              <input type="text" id="fullname" name="fullname" required>
            </div>
            <div class="form-group">
           <label for="email">Email</label>
              <input type="email" id="email" name="email" required>
            </div>
            <div class="form-group">
              <label for="message">Type Your Message</label>
              <textarea id="message" name="message" rows="4" required></textarea>
            </div>
            <button type="submit">Send</button>
            <div id="confirmation"></div>
        </form>
    </div>
    <script>
        function sendMessage() {
            let name = document.getElementById("fullname").value;
            let email = document.getElementById("email").value;
            let message = document.getElementById("message").value;
            let confirmationBox = document.getElementById("confirmation");
            confirmationBox.textContent = `Message sent successfully! `;
            confirmationBox.style.color = "white"; // ✅ Sets text color to white
console.log(`Message sent successfully!`);
           console.log(`Message sent successfully!`   );
            // Clear the form after submission
            document.getElementById("fullname").value = "";
            document.getElementById("email").value = "";
```

```
            document.getElementById("message").value = "";
        }
    </script>
</body>
</html>
```

4.2 Test Cases

4.2.1 Test case 1

Test Case ID:TC001

Test Title: Play Music

Description: Verify if the user can play a selected song.

Test steps:

1.Open the music player app

2.Select a song from the library

3.Click the “Play” button.

Expected result: The song starts playing

4.2.2 Test case2

Test case ID:TC002

Test Title: Pause music

Description: Verify if the user can pause a playing song

Test Steps:

1.play a song.

2.Click the “pause” button.

Expected result: The song pauses.

4.2.3 Test Cases 3

Test Case ID:TC003

Test Title: Skip to next Song

Description: verify if the user can skip to the next song in the playlist

Test Steps:

1.Play a song.

2.Click the “Next” button.

Expected Output: The next song in the playlist starts playing.

CHAPTER 5

RESULTS

CHAPTER-6

CONCLUSION

5.Conclusion

The ‘’RYTHM RISE’’ project successfully demonstrates the creation of a simple, visually appealing, and interactive music website using HTML and CSS. This project showcases basic web development skills, including structuring content with HTML and enhancing design with CSS. The website features a clean layout with interactive buttons and hover effects, making it user-friendly. The code is lightweight, easy to maintain, and adaptable for future enhancements. Overall, the RYTHM RISE project highlights the importance of web development fundamentals, user experience (UX) design, and code maintainability. It provides a strong foundation for further development and expansion, and potential future enhancements could include adding dynamic content, implementing responsive design, integrating social media, and expanding accessibility features.

REFERENCES

Books:

1. Duckett, J. (2019). HTML and CSS: Design and Build Websites. John Wiley & Sons.

2. W3C. (2020). HTML5: The Markup Language.

3. Meyer, E. A. (2017). CSS: The Definitive Guide.

Websites:

1. W3C (World Wide Web Consortium)

2. Mozilla Developer Network

3. WebAIM (Web Accessibility in Mind)

Online Resources:

1. Codecademy's HTML and CSS Course

2. FreeCodeCamp's Web Development Curriculum

3. W3Schools' Web Development Tutorials

Research Papers:

1. "The Impact of Web Development on Business" by Journal of Business Research.

2. "Web Accessibility and User Experience" by Journal of Accessibility and Design for All.

3. "The Future of Web Development: Trends and Challenges" by Journal of Web Engineering.

GitHub Link:

https://github.com/YaminiSarawati/music-player/tree/main/music.html