A Field Project Report on

# "MUSIC PLAYER"

**Submitted**

*In partial fulfillment of the requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE and ENGINEERING**

By

| | |
|---|---|
| G.Barna Bas | (231fa04776) |
| M.Sai Vignesh | (231fa04815) |
| G.Nandini | (231fa04858) |
| V.Keerti | (231fa04875) |

Under the Guidance of
**Dr.T.Narasimha rao**
**Assistant Professor, CSE**

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF COMPUTING AND INFORMATICS**

**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH**
**(Deemed to be University)**
**Vadlamudi, Guntur -522213, INDIA.**

**April, 2025**

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

# CERTIFICATE

This is to certify that the field project entitled "MUSIC PALYER" is being submitted by **[G. Barna Bas], [231FA04776]**, [M. Sai Vignesh], [231FA04815], [G. Nandini], [231FA04858], and [V.Keerti], [231FA04875] in partial fulfilment of the requirements for the degree of **Bachelor of Technology (B.Tech.) in Computer Science and Engineering** at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a bonafide work carried out by the aforementioned students under my guidance and supervision.

**Guide**

**Project Review Committee**

**HoD, CSE**

HoD
Dept. of Computer Science & Engineering
VFSTR Deemed to be University
VADLAMUDI - 522 213
Guntur Dist., A.P., India.

VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

## DECLARATION

Date:09-04-2025

We hereby declare that the work presented in the field project titled "MUSIC PALYER" is the result of our own efforts and investigations.

This project is being submitted under the supervision of **Dr.T.Narasimha rao, Designation** in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at the Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

| | | |
|---|---|---|
| G. Barna Bas | (231FA04776) | G. Barna Ba S |
| M. Sai Vignesh | (231FA04815) | M. Saivignesh |
| G. Nandini | (231FA04858) | G. Nandini |
| V.Keerti | (231FA04875) | V. Keerti |

# Contents

| Chapter No. | | Contents | Page No |
|---|---|---|---|
| 1 | | Introduction | 2-5 |
| | 1.1 | Problem Definition | 2 |
| | 1.2 | Existing System | 2 |
| | 1.3 | Proposed System | 3 |
| | 1.4 | Literature Review | 4 |
| 2 | | System Requirements | 5-7 |
| | 2.1 | Hardware & Software Requirements | 5 |
| | 2.2 | Software Requirements Specification(SRS) | 6-7 |
| 3 | | System Design | 7-8 |
| | 3.1 | Modules of System | 7 |
| | 3.2 | UML Diagrams | 8 |
| - 4 | | Implementation | 8-42 |
| | 4.1 | Sample Code | 8-41 |
| | 4.2 | Test Cases | 41-42 |
| 5 | | Results | 43-44 |
| | 5.1 | Output Screens | 43-44 |
| 6 | | Conclusion | 45 |
| | | References | 45 |

# 1.INTRODUCTION

Creating a web-based music player offers a wonderful opportunity to blend creativity with technical skills. As technology advances, users expect seamless, intuitive, and responsive interfaces, and a well-designed music player can greatly enhance their audio experience. This project involves integrating HTML, CSS, and JavaScript to develop a user-friendly interface with essential controls such as play, pause, volume adjustment, and track navigation.

**Key Objectives:**

1. **User Experience:** The player should be intuitive and simple to use, with clear and accessible controls.
2. **Responsiveness:** Ensure the music player looks and works well on various devices, from desktops to smartphones.
3. **Functionality:** Provide basic playback controls, volume adjustments, and the ability to navigate through audio tracks.
4. **Customization:** Allow users to personalize their experience, such as choosing themes or creating playlists.

By the end of this project, you'll have a solid understanding of web development basics and a working music player that users can enjoy. Ready to dive in? Let's get started!

## 1.1 Problem Definition:

Develop a web-based music player that allows users to play, pause, adjust volume, and navigate through audio tracks. The player should be user-friendly, responsive across devices, and provide basic controls for managing playback. The challenge lies in integrating HTML, CSS, and JavaScript to create a seamless and customizable audio experience for users.

## 1.2 Existing System:

### 1. Spotify

- **Platform**: Windows, macOS, iOS, Android, Web
- **Features**: Streaming service, music recommendations, playlists, podcasts, offline listening (premium), cross-device syncing.
- **Description**: One of the most popular music streaming platforms with a large library of songs and podcasts.

### 2. Apple Music

- **Platform**: iOS, macOS, Windows, Android, Web
- **Features**: Music streaming, curated playlists, radio stations, offline listening, integration with iTunes library.

- **Description**: Apple’s music streaming service that offers a vast collection of songs, radio stations, and curated playlists.

### 3. VLC Media Player

- **Platform**: Windows, macOS, Linux, iOS, Android
- **Features**: Plays almost any audio or video format, streaming support, customizable interface, free and open-source.
- **Description**: A versatile media player that supports a wide variety of audio and video formats.

## 1.3. Proposed System:

**System Name:**
Online Music Player Application

**Purpose:**
To allow users to search, play, and manage their favorite songs and albums through a user-friendly interface.

**Target Users:**

Music lovers

General users who want to stream and manage music

Users looking for a modern UI-based media player

**Core Features:**

🎵 Play/Pause/Skip controls

🎧 Now Playing section with song details and cover art

🔍 Search bar to find songs or albums

📁 Navigation to Home, Albums, Favorites, Recently Played, and Top Charts

♡ Like/Favorite feature

🕒 Playback progress bar with timestamp

□💻 User login/logout management

**Frontend Technologies (suggested):**

React.js or Next.js for UI rendering

Tailwind CSS or CSS Modules for styling

HTML5 <audio> element or a custom audio library (like Howler.js)

**Audio Source:**

Hosted on cloud/CDN (e.g., AWS S3, Firebase Storage)

**User Account Features (Planned/Future):**

Login/Signup system

Playlist creation

Recently played tracking

Personalized recommendations

**Admin Features (Optional):**

Upload new songs

Manage charts and categories

Moderate user content

**Device Compatibility:**

Desktop and tablet optimized

Potential for mobile responsiveness (via PWA or separate app)

## 1.4. Literature Review:

A literature survey for a music player project involves reviewing existing research,technologies,methodologies,and implementations related to music players. This survey will cover various aspects, including user interface design, audio playback technologies, and the integration of additional features. Below is a structured literature survey that can serve as a foundation for your music player project. esearch has shown that music players have had a significant impact on society, influencing culture, social interactions, and individual behavior. The music industry has been transformed, with music players changing the way music is consumed and distributed. Music players have also

influenced social interactions, enabling users to share music and connect with others. Furthermore, music players have impacted individual behavior, influencing mood, cognitive function, and physical activity.In terms of music player features, research has shown that user interface plays a crucial role in music player design. Music recommendation algorithms have also become an essential feature in music players, with research focusing on developing personalized music recommendation algorithms. Sound quality is also a critical aspect of music players, with research showing that high-quality audio can enhance the listening experience.Future research directions for music players include developing personalized music recommendation algorithms, exploring multimodal interaction techniques, and investigating the relationship between music and emotion. Additionally, research should focus on developing music players that can enhance emotional experiences and provide a more immersive listening experience.

# 2.SYSTEM REQUIREMENTS

To develop and maintain a fully functional Travel India website, it is essential to define the system requirements. These requirements include hardware, software, functional, and non-functional specifications to ensure seamless performance, security, and user experience.

## 2.1 Hardware & Software Requirements:

### Hardware

| **Category** | **Requirement** | **Minimum Specification** | **Recommended Specification** |
|---|---|---|---|
| **Client-Side Hardware** | **Processor** | Intel Core i3 (or equivalent) | Intel Core i5 or higher (or equivalent) |
| | **Memory (RAM)** | 2 GB | 4 GB or higher |
| | **Storage** | 100 MB | 500 MB or more |
| | **Display** | 1024 x 768 resolution | Full HD (1920 x 1080) or higher |
| | **Sound Card** | Required | Required |
| | **Network** | Broadband internet connection | High speed connection (100 Mbps or higher) |
| **Server-Side Hardware** (Optional) | **Processor** | Intel Xeon or equivalent (2.0 GHz) | Multi-core processor (3.0 GHz) |
| | **Memory (RAM)** | 4 GB | 8 GB or higher |
| | **Storage** | 100 GB | 1 TB or more (for large media libraries) |
| | **Network** | 100 Mbps connection | High-speed connection (1 Gbps or higher) |

## Software:

| **Category** | **Requirement** | **Details** |
|---|---|---|
| **Client-Side Software** | **Operating System** | Windows, macOS, Linux, Android, iOS |
| | **Web Browser** | Chrome, Firefox, Edge, Safari (latest versions) |
| | **Languages and Libraries** | HTML5, CSS3, JavaScript (ES6+) |
| | **Audio Format Support** | MP3, WAV, OGG, AAC |
| **Server-Side Software** (Optional) | **Backend Framework** | Node.js, PHP, Python, Java |
| | **Database** | MySQL, MongoDB (for playlists and user data) |
| | **API** | RESTful API (for communication between frontend and backend) |
| **Development Tools** | **Code Editor** | Visual Studio Code, Sublime Text, Atom |
| | **Version Control** | Git, GitHub, GitLab |
| | **Testing Tools** | Chrome DevTools, Postman |

## 2.2 Software Requirements Specification (SRS) for Music Player:

### 1. **Introduction**

The Music Player application is designed to offer users a seamless experience for playing, organizing, and managing audio files. It aims to provide high-quality playback, intuitive navigation, and customization options to enhance user satisfaction.

### 2. **System Features**

### Functional Requirements:

- **Audio Playback**: Support for multiple audio file formats (e.g., MP3, WAV, AAC).
- **Playlist Management**: Create, edit, and manage playlists.
- **Search Functionality**: Ability to search for songs, artists, or albums.
- **Shuffle and Repeat Modes**: Options for playback customization.

**Equalizer Settings**: Adjustable audio settings for improved sound quality.

## Non-functional Requirements:

- **Performance**: Smooth playback with minimal CPU and memory usage.
- **User Interface**: Intuitive and visually appealing design for easy navigation.
- **Compatibility**: Cross-platform support (Windows, macOS, iOS, Android).
- **Security**: Protection of user data, including playlists and preferences.

## 3. **Hardware Requirements**

- Minimum 2 GB RAM.
- Processor: 1 GHz or higher.
- Storage: 100 MB of free space for application installation.

## 4. **Constraints**

- Internet connection required for streaming services.
- Limited to audio files; does not support video playback

# 3.SYSTEM DESIGN

## 3.1 Modules of System:

| Module | Purpose | Technologies |
|---|---|---|
| **User Interface** | Manage player UI and user interaction | HTML, CSS, JavaScript |
| **Audio Control** | Handle audio playback and controls | HTML5 Audio API, JavaScript |
| **Playlist Management** | Manage playlist creation and storage | JavaScript, Node.js, MySQL/MongoDB |
| **Track Management** | Manage track metadata and retrieval | File system, Cloud storage |
| **User Authentication** | Manage user login and security | OAuth, JWT |
| **Backend API** | Handle data exchange between frontend and backend | Node.js, PHP, Python |
| **Database** | Store user and track data | MySQL, MongoDB |
| **Error Handling** | Manage and log system errors | JavaScript, Node.js |
| **Settings and Preferences** | Customize player settings and UI | JavaScript, Local Storage |

## 3.2 UML Diagrams

```
Root Node
FP101
- -  Folders
-  -  -   - - - images
-  -  -   - - - songs
- -   Files
-  -  -   - - -  i.css
-  -  -   - - - i.html
-  -  -   - - - i.js
-  -  -   - - - i3.css
-  -  -   - - - i3.html
-  -  -   - - - i3.j
```

# 4. IMPLEMENTATION

## 4.1 Sample Code:

**Homepage html(i3.html):**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Melodify - Your Music Companion</title>
    <link rel="stylesheet" href="i3.css">
</head>
<body>
    <div class="container">
        <!-- Background animated circles -->
        <div class="background">
            <div class="blob blob-1"></div>
            <div class="blob blob-2"></div>
            <div class="blob blob-3"></div>
        </div>

        <!-- Main content -->
        <div class="content">
            <!-- Logo -->
            <div class="logo">
                <svg xmlns="http://www.w3.org/2000/svg" width="84" height="84" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="logo-icon">
                    <path d="M8 19V5L19 4v13"></path>
                    <circle cx="4" cy="18" r="2"></circle>
                    <circle cx="15" cy="17" r="2"></circle>
                </svg>
                <svg xmlns="http://www.w3.org/2000/svg" width="84" height="84" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="logo-icon-ping">
                    <path d="M8 19V5L19 4v13"></path>
                    <circle cx="4" cy="18" r="2"></circle>
                    <circle cx="15" cy="17" r="2"></circle>
                </svg>
            </div>

            <!-- App name and tagline -->
            <div class="app-info">
                <h1>Music</h1>
                <p>Your personal symphony of endless musical possibilities</p>
            </div>

            <!-- CTA Button -->
            <button id="startListening" class="cta-button">
                <span class="button-content">
                    Start Listening
                    <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20"
viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-
linecap="round" stroke-linejoin="round">
                        <polygon points="5 3 19 12 5 21 5 3"></polygon>
                    </svg>
                </span>
            </button>

            <!-- Features -->

    <script src="i3.js"></script>
</body>
</html>
```

**Home page css(i3.css):**

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
     font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica
Neue", Arial, sans-serif;
    min-height: 100vh;
    color: white;
    overflow-x: hidden;
}

.container {
    min-height: 100vh;
    background: linear-gradient(135deg, #4c1d95 0%, #000000 50%, #000000 100%);
    position: relative;
}

/* Background blobs */
.background {
    position: absolute;
    inset: 0;
    overflow: hidden;
}

.blob {
    position: absolute;
    width: 18rem;
    height: 18rem;
    border-radius: 50%;
    filter: blur(40px);
    mix-blend-mode: multiply;
    opacity: 0.2;
    animation: blob 7s infinite;
}

.blob-1 {
    background: #9333ea;
    top: 25%;
    left: -1rem;
}

.blob-2 {
    background: #2563eb;
    top: 25%;
    right: -1rem;
    animation-delay: 2s;
}
```

```
.blob-3 {
    background: #db2777;
    bottom: -2rem;
    left: 50%;
    transform: translateX(-50%);
    animation-delay: 4s;
}

/* Main content */
.content {
    position: relative;
    min-height: 100vh;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    padding: 1rem;
    text-align: center;
}

/* Logo */
.logo {
    position: relative;
    margin-bottom: 2rem;
}


.
@media (min-width: 768px) {
    .features {
        grid-template-columns: repeat(3, 1fr);
    }

    h1 {
        font-size: 4rem;
    }
}
```

**Home page js(i3.js):**

```
document.addEventListener('DOMContentLoaded', () => {
    const startListeningButton = document.getElementById('startListening');

    startListeningButton.addEventListener('click', () => {
        // This will be implemented later for navigation
       window.location.href='i.html';
    });
});
```

**MusicPlayer html(i.html):**

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Music Player</title>
    <link rel="stylesheet" href="i.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="icon" href="images/top-charts.jpg"> <!-- Placeholder image -->
</head>
<body>
    <div class="app-container">
        <!-- Sidebar -->
        <div class="sidebar">
            <div class="logo">
                <i class="fas fa-music"></i>
                <h2> Music Player</h2>
            </div>
            <!-- Add search bar -->
            <div class="search-container">
                <input type="text" id="search-input" placeholder="🔍 Search songs or albums...">
                <i class="fas fa-search search-icon"></i>
            </div>
            <nav>
                <ul>
                    <li class="active" data-view="home"><i class="fas fa-home"></i> 🏠 Home</li>
                    <li data-view="albums"><i class="fas fa-compact-disc"></i> ⦿ Albums</li>
                    <li data-view="favorites"><i class="fas fa-heart"></i> ♡ Favorites</li>
                    <li data-view="search-results" class="hidden"><i class="fas fa-search"></i> 🔍 Search Results</li>
                    <li data-view="recently-played"><i class="fas fa-history"></i> 🕒 Recently Played</li>
                    <li data-view="top-charts"><i class="fas fa-chart-line"></i> 📈 Top Charts</li>
                </ul>
            </nav>
            <div class="user-section">
                <div class="user-icon" id="user-icon">
                    <i class="fas fa-user"></i>
                </div>
                <div class="login-status" id="login-status">
                    <span>👤 Not logged in</span>
                </div>
                <!-- Add logout button -->
                <button id="logout-btn" class="logout-btn hidden">
                    <i class="fas fa-sign-out-alt"></i> 🚪
                </button>
            </div>
        </div>

        <!-- Main Content -->
        <div class="main-content">
            <!-- Login Modal -->
            <div class="login-modal" id="login-modal">
                <div class="login-form">
                    <h2>🔑 Login</h2>
                    <div class="close-modal"><i class="fas fa-times"></i></div>
                    <form id="login-form">
                        <div class="form-group">
                            <label for="username">👤 Username</label>
                            <input type="text" id="username" required>
                        </div>
                        <div class="form-group">
                            <label for="password">🔒 Password</label>
                            <input type="password" id="password" required>
                        </div>
            <button type="submit" class="login-btn">🔒 Login</button>
          </form>
        </div>
      </div>

      <!-- Search Results View -->
      <div class="view hidden" id="search-results-view">
        <h1>🔍 Search Results</h1>
        <div class="search-results-container">
          <div class="search-section">
            <h2>🎧 Songs</h2>
            <ul id="song-results" class="search-results-list"></ul>
          </div>
          <div class="search-section">
            <h2>💿 Albums</h2>
            <div id="album-results" class="albums-grid"></div>
          </div>
        </div>
      </div>

      <!-- Home View -->
      <div class="view" id="home-view">
        <h1>🎧 Now Playing</h1>
        <div class="now-playing">
          <div class="album-cover">
             <img id="current-album-cover" src="https://images.unsplash.com/photo-
1470225620780-dba8ba36b745?ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D
&auto=format&fit=crop&w=1470&q=80" alt="Album Cover">
          </div>
          <div class="song-details">
            <h2 id="current-song-title">Ayudha Pooja</h2>
            <p id="current-song-artist">🎤 Anirudh Ravichander</p>
            <p id="current-album-title">💿 Devara</p>
          </div>
        </div>

        <div class="player-controls">
        <div class="progress-container">
            <div class="progress-bar">
                <div class="progress" id="progress"></div>
            </div>
            <div class="time">
                <span id="current-time">⏱ 0:00</span>
                <span id="duration">⏱ 3:30</span>
            </div>
        </div>

        <div class="controls">
            <button id="prev-btn"><i class="fas fa-backward"></i> </button>
            <button id="play-btn"><i class="fas fa-play"></i> </button>
            <button id="next-btn"><i class="fas fa-forward"></i> </button>
            <button id="favorite-btn"><i class="far fa-heart"></i> ❤</button>
        </div>
    </div>

    <div class="current-playlist">
        <h3>📜 Current Album</h3>
        <ul id="current-playlist"></ul>
    </div>
</div>

<!-- Albums View -->
<div class="view hidden" id="albums-view">
    <h1>💿 Albums</h1>
    <div class="albums-grid" id="albums-grid"></div>
</div>

<!-- Favorites View -->
<div class="view hidden" id="favorites-view">
    <h1>❤ Favorites</h1>
    <ul id="favorites-list" class="favorites-list"></ul>
</div>

<!-- Recently Played View -->
<div class="view hidden" id="recently-played-view">
```

```
                <h1>🕒 Recently Played</h1>
                <ul id="recently-played-list" class="recently-played-list"></ul>
            </div>

            <!-- Top Charts View -->
            <div class="view hidden" id="top-charts-view">
                <h1>📈 Top Charts</h1>
                <ul id="top-charts-list" class="top-charts-list"></ul>
            </div>
        </div>
    </div>

    <!-- Audio Elements -->
    <audio id="audio-player"></audio>

    <script src="i.js"></script>
</body>
</html>
```

**Music Playercss(i.css):**

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

@keyframes gradientBG {
    0%, 100% { background-position: 0% 50%; }
    50% { background-position: 100% 50%; }
}

@keyframes shine {
    0% { transform: translateX(-100%); }
    100% { transform: translateX(100%); }
}

@keyframes pulse {
    0%, 100% { transform: scale(1); }
    50% { transform: scale(1.05); }
}
```

```
body {
    font-family: 'Arial', sans-serif;
    background: linear-gradient(-45deg, #1a1a1a, #2d1b31, #1f1f3a, #2d1b31);
    background-size: 400% 400%;
    animation: gradientBG 15s ease infinite;
    color: white;
    min-height: 100vh;
    overflow-x: hidden;
    position: relative;
}

body::before {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: url('images/music-notes-bg.png') center/cover no-repeat;
    opacity: 0.05; /* Subtle music notes background */
    z-index: -1;
}

body::after {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: radial-gradient(circle, rgba(255, 255, 255, 0.1), transparent 70%);
    z-index: -1;
    pointer-events: none;
}

.app-container {
    display: flex;
    min-height: 100vh;
```

```
}

/* Sidebar Styles */
.sidebar {
    width: 250px;
    background: rgba(0, 0, 0, 0.7);
    backdrop-filter: blur(10px);
    padding: 20px;
    display: flex;
    flex-direction: column;
    position: fixed;
    height: 100vh;
    z-index: 10;
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.5);
}

.logo {
    display: flex;
    align-items: center;
    gap: 10px;
    margin-bottom: 40px;
    padding-bottom: 20px;
    border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}

.logo i {
    font-size: 24px;
    color: #646cff;
}

nav ul {
    list-style: none;
}

nav ul li {
    padding: 12px 15px;
    margin-bottom: 5px;
    border-radius: 8px;
    cursor: pointer;
    transition: all 0.3s ease;
    display: flex;
    align-items: center;
    gap: 10px;
}

nav ul li:hover {
    background: rgba(255, 255, 255, 0.1);
}

nav ul li.active {
    background: rgba(100, 108, 255, 0.2);
    color: #646cff;
}

nav ul li i {
    width: 20px;
    text-align: center;
}

.user-section {
    margin-top: auto;
    padding-top: 20px;
    border-top: 1px solid rgba(255, 255, 255, 0.1);
    display: flex;
    align-items: center;
    gap: 10px;
}

.user-icon {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    background: #646cff;
    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    transition: all 0.3s ease;
    box-shadow: 0 0 10px rgba(100, 108, 255, 0.5);
}

.user-icon:hover {
    transform: scale(1.05);
    box-shadow: 0 0 20px rgba(100, 108, 255, 0.8);
}

.login-status {
    font-size: 14px;
    color: rgba(255, 255, 255, 0.7);
}

/* Search Container */
.search-container {
    position: relative;
    margin: 0 10px 20px;
    padding: 0 5px;
}

.search-container input {
    width: 100%;
    padding: 10px 30px 10px;
    background: rgba(255, 255, 255, 0.1);
    border: 1px solid rgba(255, 255, 255, 0.2);
    border-radius: 8px;
    color: white;
    font-size: 14px;
    transition: all 0.3s ease;
}

.search-container input:focus {
    outline: none;
    border-color: #646cff;
    background: rgba(255, 255, 255, 0.15);
}

.search-container .search-icon {
    position: absolute;
    right: 15px;
    top: 50%;
    transform: translateY(-50%);
    color: rgba(255, 255, 255, 0.7);
    pointer-events: none;
}

/* Logout Button */
.logout-btn {
    background: none;
    border: none;
    color: rgba(255, 255, 255, 0.7);
    cursor: pointer;
    padding: 5px;
    margin-left: 10px;
    border-radius: 50%;
    transition: all 0.3s ease;
    box-shadow: 0 0 10px rgba(100, 108, 255, 0.5);
}

.logout-btn:hover {
    color: white;
    background: rgba(255, 255, 255, 0.1);
    box-shadow: 0 0 20px rgba(100, 108, 255, 0.8);
}

/* Main Content Styles */
.main-content {
    flex: 1;
    padding: 30px;
    margin-left: 250px;
    position: relative;
}

.view {
    animation: fadeIn 0.5s ease;
}

@keyframes fadeIn {
    from { opacity: 0; }
    to { opacity: 1; }
}

.hidden {
    display: none !important;
}

h1 {
    font-size: 32px;
    margin-bottom: 30px;
    position: relative;
    display: inline-block;
}

h1::after {
    content: '';
    position: absolute;
    bottom: -10px;
    left: 0;
    width: 50px;
    height: 3px;
    background: #646cff;
    border-radius: 3px;
}

/* Now Playing Section */
.now-playing {
    display: flex;
    gap: 30px;
    margin-bottom: 40px;
    background: rgba(255, 255, 255, 0.05);
    border-radius: 15px;
    padding: 30px;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
}

.album-cover {
    width: 250px;
    height: 250px;
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.3);
    transition: transform 0.3s ease;
}

.album-cover:hover {
    transform: scale(1.02);
}

.album-cover img {
    width: 100%;
    height: 100%;
    object-fit: cover;
}

.song-details {
    display: flex;
    flex-direction: column;
    justify-content: center;
}

.song-details h2 {
    font-size: 28px;
    margin-bottom: 10px;
}

.song-details p {
    color: rgba(255, 255, 255, 0.7);
    margin-bottom: 5px;
}

/* Player Controls */
.player-controls {
    background: rgba(255, 255, 255, 0.1);
    backdrop-filter: blur(10px);
    padding: 25px;
    border-radius: 15px;
    margin-bottom: 40px;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
}

.progress-container {
    margin-bottom: 20px;
}

/* Enhanced Progress Bar */
.progress-bar {
    width: 100%;
    height: 8px; /* Slightly thicker for better visibility */
    background: rgba(255, 255, 255, 0.2);
    border-radius: 10px;
    margin-bottom: 10px;
    cursor: pointer;
    position: relative;
    overflow: hidden;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

.progress {
    width: 0%;
    height: 100%;
    background: linear-gradient(90deg, #ff4d6d, #646cff, #535bf2);
    border-radius: 10px;
    transition: width 0.1s linear;
    box-shadow: 0 0 15px rgba(100, 108, 255, 0.7), 0 0 25px rgba(255, 77, 109, 0.5);
    position: relative;
}

.progress::after {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.3), transparent);
    animation: shine 2s infinite;
    border-radius: 10px;
}

.time {
    display: flex;
    justify-content: space-between;
    color: rgba(255, 255, 255, 0.7);
    font-size: 14px;
}

.controls {
    display: flex;
    justify-content: center;
    align-items: center;
    gap: 20px;
}

.controls button {
    background: none;
    border: none;
    color: white;
    font-size: 18px;
    cursor: pointer;
    width: 50px;
    height: 50px;
    border-radius: 50%;

    }

    .now-playing {
        flex-direction: column;
        align-items: center;
        text-align: center;
    }

    .album-cover {
        width: 200px;
        height: 200px;
    }

    .albums-grid {
        grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
    }
}
```

**Music Playerjs(i.js):**

```
document.addEventListener('DOMContentLoaded', () => {
    // DOM Elements
    const audioPlayer = document.getElementById('audio-player');
    const playBtn = document.getElementById('play-btn');
    const prevBtn = document.getElementById('prev-btn');
    const nextBtn = document.getElementById('next-btn');
    const favoriteBtn = document.getElementById('favorite-btn');
    const progress = document.getElementById('progress');
    const progressBar = document.querySelector('.progress-bar');
    const currentTimeEl = document.getElementById('current-time');
    const durationEl = document.getElementById('duration');
    const currentSongTitle = document.getElementById('current-song-title');
    const currentSongArtist = document.getElementById('current-song-artist');
    const currentAlbumTitle = document.getElementById('current-album-title');
    const currentAlbumCover = document.getElementById('current-album-cover');
    const currentPlaylist = document.getElementById('current-playlist');
    const albumsGrid = document.getElementById('albums-grid');
    const favoritesList = document.getElementById('favorites-list');
    const navItems = document.querySelectorAll('nav ul li');
    const views = document.querySelectorAll('.view');
    const userIcon = document.getElementById('user-icon');
    const loginModal = document.getElementById('login-modal');
    const closeModal = document.querySelector('.close-modal');
    const loginForm = document.getElementById('login-form');
    const loginStatus = document.getElementById('login-status');

    // Music data
    const albums = [
        {
            id: 'album1',
            title: 'Devara',
            artist: 'Thaman S',
            coverUrl: 'images/de.jpg',
            songs: [
                { id: 'song1', title: 'Ayudha Pooja', artist: 'Anirudh Ravichander', duration: 210,
url: 'songs/Ayudha Pooja.mp3' },
             { id: 'song2', title: 'Daavudi', artist: 'Anirudh Ravichander', duration: 195, url:
'songs/Daavudi.mp3' },
               { id: 'song3', title: 'Fear', artist: 'Anirudh Ravichander', duration: 180, url:
'songs/Fear.mp3' },
             { id: 'song7', title: 'Red Sea', artist: 'Anirudh Ravichander', duration: 215, url:
'songs/Red Sea.mp3' },
                { id: 'song8', title: 'All Hail The Tiger', artist: 'Anirudh Ravichander', duration:
205, url: 'songs/All Hail The Tiger.mp3' },
           { id: 'song9', title: 'Chuttamalle', artist: 'Anirudh Ravichander', duration: 190, url:
'songs/Chuttamalle.mp3' }
            ]
        },
        {
            id: 'album2',
            title: 'hi-NANNA',
            artist: 'Hashim',
            coverUrl: 'images/hi.jpg',
            songs: [
                     { id: 'song4', title: 'Odiyamma', artist: 'Hashim', duration: 225, url:
'songs/Odiyamma.mp3' },
                     { id: 'song5', title: 'Samayama', artist: 'Hashim', duration: 198, url:
'songs/Samayama.mp3' },
          { id: 'song6', title: 'Gaaju Bomma', artist: 'Hashim', duration: 187, url: 'songs/Gaaju
Bomma.mp3' }
            ]
        },
        {
            id: 'album3',
            title: 'Bramhastra',
            artist: 'Arjith singh',
            coverUrl: 'images/bram.jpg',
            songs: [
                  { id: 'song1', title: 'Kumkumala', artist: 'Arjith singh', duration: 210, url:
'songs/bramhastram/Kumkumala.mp3' },
                     { id: 'song2', title: 'Deva', artist: 'Arjith singh', duration: 195, url:
```

```
'songs/bramhastram/Deva Deva.mp3' },
                { id: 'song3', title: 'Allari Motha', artist: 'Arjith singh', duration: 180, url:
'songs/bramhastram/Allari Motha.mp3' }
          ]
      },
      {
          id: 'album4',
          title: 'Return of Dragon',
          artist: 'Sarath Santhosh',
          coverUrl: 'images/drag.jpg',
          songs: [
              { id: 'song1', title: 'Madhuvaramae', artist: 'Sarath Santhosh', duration: 210, url:
'songs/dragon/madhu.mp3' },
              { id: 'song2', title: 'Rise of Dragon', artist: 'Sarath Santhosh', duration: 195, url:
'songs/dragon/rise of.mp3' },
                { id: 'song3', title: 'Yendhukae', artist: 'Sarath Santhosh', duration: 180, url:
'songs/dragon/yendhuke.mp3' }
          ]
      },
      {
          id: 'album5',
          title: 'Vikram-Hitlist',
          artist: 'Anirudh Ravichandran',
          coverUrl: 'images/vik.jpg',
          songs: [
              { id: 'song1', title: 'Vikram title song', artist: 'Anirudh Ravichander', duration:
210, url: 'songs/vikram/title.mp3' },
              { id: 'song2', title: 'once upon a time', artist: 'Anirudh Ravichander', duration:
195, url: 'songs/vikram/once.mp3' },
             { id: 'song3', title: 'Porata Simham', artist: 'Anirudh Ravichander', duration: 180,
url: 'songs/vikram/porata.mp3' }
          ]
      },

   {
      id: 'album6',
      title: 'Orange',
      artist: 'Harris Jayaraj',
      coverUrl: 'images/orange.jpg',
      songs: [
                { id: 'song1', title: 'Rooba', artist: 'Karthik', duration: 240, url:
'songs/orange/rooba.mp3' },
            { id: 'song2', title: 'Nenu Nuvvantu', artist: 'Chinmayi', duration: 210, url:
'songs/orange/ola.mp3' },
            { id: 'song3', title: 'Hello Rammante', artist: 'Karunya', duration: 200, url:
'songs/orange/hello.mp3' }
      ]
   },
   {
      id: 'album7',
      title: 'Ambajipetta',
      artist: 'Keeravani',
      coverUrl: 'images/ambaji.jpg',
      songs: [
           { id: 'song1', title: 'Bhajan', artist: 'SP Balasubrahmanyam', duration: 220, url:
'songs/ambaji/gumma.mp3' },
      ]
   },
   {
      id: 'album8',
      title: 'Arya',
      artist: 'Devi Sri Prasad',
      coverUrl: 'images/arya.jpg',
      songs: [
        { id: 'song1', title: 'Feel My Love', artist: 'Karthik', duration: 230, url: 'songs/arya/feel
my.mp3' },

      ]
   },
   {
      id: 'album9',
      title: 'Bommarillu',
      artist: 'Mickey J Meyer',
      coverUrl: 'images/bomma.jpg',
      songs: [
             { id: 'song1', title: 'Bomma', artist: 'Armaan Malik', duration: 240, url:
'songs/bomma/appudo.mp3' },
      ]
    },
    {
      id: 'album10',
      title: 'Geetha Govindham',
      artist: 'Gopi Sundar',
      coverUrl: 'images/geetha.jpg',
      songs: [
              { id: 'song1', title: 'Inkem', artist: 'Sid Sriram', duration: 250, url:
  'songs/geetha/inkem.mp3' },
             { id: 'song2', title: 'Vachindamma', artist: 'Saketh', duration: 230, url:
  'songs/geetha/vachindhamma.mp3' },
              { id: 'song3', title: 'Yenti', artist: 'Chinmayi', duration: 240, url:
  'songs/geetha/yenti_yenti.mp3' }
      ]
    },
    {
      id: 'album11',
      title: 'Guntur',
      artist: 'S. Thaman',
      coverUrl: 'images/guntur.jpg',
      songs: [
          { id: 'song1', title: 'dummasala biryani', artist: 'Kailash Kher', duration: 260, url:
  'songs/guntur/dum.mp3' },
            { id: 'song2', title: 'kurchini', artist: 'Anurag Kulkarni', duration: 240, url:
  'songs/guntur/kurchi.mp3' },
             { id: 'song3', title: 'Nee Kanti', artist: 'Sid Sriram', duration: 250, url:
  'songs/guntur/nee_kanti.mp3' }
      ]
    },
    {
      id: 'album12',
      title: 'Krishnarjunayudham',
      artist: 'Chakri',
      coverUrl: 'images/krishna.jpg',
      songs: [
             { id: 'song1', title: 'FLY ', artist: 'Vijay Yesudas', duration: 240, url:
  'songs/krishna/fly.mp3' },
              { id: 'song2', title: 'MNASE', artist: 'Karthik', duration: 230, url:
```

```
'songs/krishna/manase.mp3' },

        ]
    },
    {
        id: 'album13',
        title: 'Master',
        artist: 'Anirudh Ravichander',
        coverUrl: 'images/master.jpg',
        songs: [
            { id: 'song1', title: 'Vaathi Coming', artist: 'Anirudh Ravichander', duration: 210,
url: 'songs/master/vaathi_coming.mp3' },
            { id: 'song2', title: 'Kutti Story', artist: 'Vijay', duration: 200, url:
'songs/master/kutti_story.mp3' }
        ]
    },
    {
        id: 'album14',
        title: 'Mirchi',
        artist: 'Devi Sri Prasad',
        coverUrl: 'images/mirchi.jpg',
        songs: [
            { id: 'song1', title: 'Mirchi Title Song', artist: 'Chinmayi', duration: 220, url:
'songs/mirchi/title_song.mp3' },
            { id: 'song2', title: 'Idhedho Bagundhe', artist: 'Vijay Prakash', duration: 230, url:
'songs/mirchi/idhedho_bagundhe.mp3' }
        ]
    },
    {
        id: 'album15',
        title: 'Nani Gang',
        artist: 'Gopi Sundar',
        coverUrl: 'images/nanigang.jpg',
        songs: [
            { id: 'song1', title: 'Gang Leader Title Song', artist: 'Anirudh Ravichander', duration:
240, url: 'songs/nanigang/title_song.mp3' },
            { id: 'song2', title: 'Hoyna', artist: 'Inno Genga', duration: 220, url:
'songs/nanigang/hoyna_hoyna.mp3' }
        ]
    },
    {
      id: 'album16',
      title: 'Nannaku Prematho',
      artist: 'Devi Sri Prasad',
      coverUrl: 'images/nannaku.jpg',
      songs: [
              { id: 'song1', title: 'Follow', artist: 'Jr. NTR', duration: 210, url:
'songs/nannaku/follow_follow.mp3' },
          { id: 'song2', title: 'Love Dhebba', artist: 'Sooraj Santhosh', duration: 230, url:
'songs/nannaku/love_dhebba.mp3' }
      ]
    },
    {
      id: 'album17',
      title: 'Ori',
      artist: 'Thaman S',
      coverUrl: 'images/ori.jpg',
      songs: [
            { id: 'song1', title: 'Ori Devuda', artist: 'Sid Sriram', duration: 250, url:
'songs/ori/ori_devuda.mp3' },
            { id: 'song2', title: 'Avunanavaa', artist: 'Haricharan', duration: 240, url:
'songs/ori/avunanavaa.mp3' }
      ]
    },
    {
      id: 'album18',
      title: 'RRR',
      artist: 'M. M. Keeravani',
      coverUrl: 'images/rrr.jpg',
      songs: [
            { id: 'song1', title: 'Naatu', artist: 'Rahul Sipligunj', duration: 260, url:
'songs/rrr/naatu_naatu.mp3' },
             { id: 'song2', title: 'Dosti', artist: 'Amit Trivedi', duration: 240, url:
'songs/rrr/dosti.mp3' }
      ]
    },
    {
      id: 'album19',
      title: 'Taxi',
      artist: 'Gopi Sundar',
      coverUrl: 'images/taxi.jpg',
      songs: [
          { id: 'song1', title: 'Taxi Title Song', artist: 'Armaan Malik', duration: 220, url: 'songs/taxi/title_song.mp3' },
          { id: 'song2', title: 'Neeve', artist: 'Sid Sriram', duration: 230, url: 'songs/taxi/neeve_neeve.mp3' }
      ]
    },
    {
      id: 'album20',
      title: 'V',
      artist: 'Amit Trivedi',
      coverUrl: 'images/v.jpg',
      songs: [
          { id: 'song1', title: 'Manasu Maree', artist: 'Shashaa Tirupati', duration: 240, url: 'songs/v/manasu_maree.mp3' },
          { id: 'song2', title: 'Vishwaroopam', artist: 'Anirudh Ravichander', duration: 250, url: 'songs/v/vishwaroopam.mp3' }
      ]
    }
    ];

    // App state
    let currentAlbum = albums[0];
    let currentSongIndex = 0;
    let isPlaying = false;
    let favorites = [];
    let isLoggedIn = false;
    let username = ";
    let recentlyPlayed = [];

    const topCharts = [
      { title: 'Daavudi', artist: 'Ed Sheeran', album: 'Devara', url: 'songs/Daavudi.mp3', coverUrl: 'images/de.jpg' },
      { title: 'Samayama', artist: 'chakri', album: 'After Hours', url: 'songs/Samayama.mp3', coverUrl: 'images/hi.jpg' },
      { title: 'inkem', artist: 'chakri', album: 'Future Nostalgia', url: 'songs/inkem.mp3',
coverUrl: 'images/geetha.jpg' },
        { title: 'Taxiwala', artist: 'Silpa', album: 'Stay', url: 'songs/maate.mp3', coverUrl:
'images/taxi.jpg' },
          { title: 'RRR', artist: 'DSP', album: 'RRR', url: 'songs/natu.mp3', coverUrl:
'images/rrr.jpg' },
   ];

   // Initialize the app
   function init() {
      loadAlbum(currentAlbum);
      renderAlbums();
      setupEventListeners();
      renderTopCharts(); // Initialize the top charts
   }
   // Load album and its songs
   function loadAlbum(album) {
      currentAlbum = album;
      currentSongIndex = 0;

      // Update UI
      currentAlbumTitle.textContent = album.title;
      currentAlbumCover.src = album.coverUrl;

      // Load first song
      loadSong(album.songs[0]);

      // Render playlist
      renderPlaylist();
   }

   // Load a specific song
   function loadSong(song, album = null) {
      currentSongTitle.textContent = song.title;
      currentSongArtist.textContent = song.artist;

      // Set the audio source
      audioPlayer.src = song.url;

      // Update UI
```

```
    updateFavoriteButton();
    resetProgress();

    if (album) {
        currentAlbum = album; // Update current album if provided
        currentAlbumTitle.textContent = album.title;
        currentAlbumCover.src = album.coverUrl;
    }

    if (isPlaying) {
        playAudio();
    }
}
// Play the current song
function playAudio() {
    const currentSong = currentAlbum.songs[currentSongIndex];
    addToRecentlyPlayed(currentSong, currentAlbum);

    audioPlayer.play().catch(error => {
        console.error("Error playing audio:", error);
        startProgressSimulation();
    });

    playBtn.innerHTML = '<i class="fas fa-pause"></i>';
    playBtn.classList.add('playing');
    isPlaying = true;
}

// Pause the current song
function pauseAudio() {
    audioPlayer.pause();

    playBtn.innerHTML = '<i class="fas fa-play"></i>';
    playBtn.classList.remove('playing');
    isPlaying = false;

    stopProgressSimulation();
}
```

```
  // Play next song
  function playNext() {
    currentSongIndex = (currentSongIndex + 1) % currentAlbum.songs.length;
    loadSong(currentAlbum.songs[currentSongIndex]);
    highlightCurrentSong();

    if (isPlaying) {
      playAudio();
    }
  }

  // Play previous song
  function playPrev() {
    currentSongIndex = (currentSongIndex - 1 + currentAlbum.songs.length) % currentAlbum.songs.length;
    loadSong(currentAlbum.songs[currentSongIndex]);
    highlightCurrentSong();

    if (isPlaying) {
      playAudio();
    }
  }


  // Toggle favorite status for current song or specific song
  function toggleFavorite(songId) {
    const id = songId || currentAlbum.songs[currentSongIndex].id;

    if (favorites.includes(id)) {
      favorites = favorites.filter(fid => fid !== id);
    } else {
      favorites.push(id);
    }

    updateFavoriteButton(); // Update the favorite button for the current song
    renderFavorites(); // Re-render the favorites list
  }

  // Update favorite button based on current song
```

```
function updateFavoriteButton() {
    const currentSong = currentAlbum.songs[currentSongIndex];

    if (favorites.includes(currentSong.id)) {
        favoriteBtn.innerHTML = '<i class="fas fa-heart"></i>';
        favoriteBtn.classList.add('active');
    } else {
        favoriteBtn.innerHTML = '<i class="far fa-heart"></i>';
        favoriteBtn.classList.remove('active');
    }
}

// Render the current playlist
function renderPlaylist() {
    currentPlaylist.innerHTML = '';

    currentAlbum.songs.forEach((song, index) => {
        const li = document.createElement('li');
        li.className = index === currentSongIndex ? 'active' : '';
        li.dataset.index = index;

        li.innerHTML = `
            <div class="song-info">
                <span class="song-number">${index + 1}</span>
                <div class="song-title-artist">
                    <h4>${song.title}</h4>
                    <p>${song.artist}</p>
                </div>
            </div>
            <div class="song-actions">
               <button class="favorite-song ${favorites.includes(song.id) ? 'active' : ''}" data-id="${song.id}">
                    <i class="${favorites.includes(song.id) ? 'fas' : 'far'} fa-heart"></i>
                </button>
            </div>
        `;

        currentPlaylist.appendChild(li);
    });
```

```
  }

  // Render all albums
  function renderAlbums() {
    albumsGrid.innerHTML = '';

    albums.forEach(album => {
      const albumCard = document.createElement('div');
      albumCard.className = 'album-card';
      albumCard.dataset.id = album.id;

      albumCard.innerHTML = `
        <img src="${album.coverUrl}" alt="${album.title}">
        <div class="play-overlay"><i class="fas fa-play"></i></div>
        <div class="album-info">
          <h3>${album.title}</h3>
          <p>${album.artist}</p>
        </div>
      `;

      albumsGrid.appendChild(albumCard);
    });
  }

  // Render favorites list
  function renderFavorites() {
    favoritesList.innerHTML = '';

    if (favorites.length === 0) {
      favoritesList.innerHTML = '<p class="text-center p-4">No favorites yet. Add some songs to your favorites!</p>';
      return;
    }

    const favoriteSongs = albums.flatMap(album =>
      album.songs.filter(song => favorites.includes(song.id)).map(song => ({
        ...song,
        albumTitle: album.title,
        albumId: album.id
```

```
            }))
        );

        favoriteSongs.forEach((song, index) => {
            const li = document.createElement('li');

            li.innerHTML = `
                <div class="song-info">
                    <span class="song-number">${index + 1}</span>
                    <div class="song-title-artist">
                        <h4>${song.title}</h4>
                        <p>${song.artist} • ${song.albumTitle}</p>
                    </div>
                </div>
                <div class="song-actions">
                    <button class="play-favorite" data-album="${song.albumId}" data-song="${song.id}">
                        <i class="fas fa-play"></i>
                    </button>
                    <button class="remove-favorite" data-id="${song.id}">
                        <i class="fas fa-times"></i>
                    </button>
                </div>
            `;

            favoritesList.appendChild(li);
        });
    }


    // Search functionality
    function setupSearch() {
        const searchInput = document.getElementById('search-input');
        let searchTimeout;

        searchInput.addEventListener('input', (e) => {
            clearTimeout(searchTimeout);
            searchTimeout = setTimeout(() => {
                const query = e.target.value.toLowerCase().trim();
            if (query.length >= 2) {
                performSearch(query);
                switchView('search-results');
            }
        }, 300);
    });
}

function performSearch(query) {
    const songResults = document.getElementById('song-results');
    const albumResults = document.getElementById('album-results');

    const matchingSongs = [];
    const matchingAlbums = [];

    albums.forEach(album => {
        if (album.title.toLowerCase().includes(query) ||
            album.artist.toLowerCase().includes(query)) {
            matchingAlbums.push(album);
        }

        album.songs.forEach(song => {
            if (song.title.toLowerCase().includes(query) ||
                song.artist.toLowerCase().includes(query)) {
                matchingSongs.push({...song, albumTitle: album.title, albumId: album.id});
            }
        });
}

// Helper function to get current search results

function getCurrentSearchResults() {

    const query = document.getElementById('search-input').value.toLowerCase().trim();

    const results = [];

    albums.forEach(album => {
          album.songs.forEach(song => {

              if (song.title.toLowerCase().includes(query) ||

                  song.artist.toLowerCase().includes(query)) {

                  results.push({...song, albumTitle: album.title, albumId: album.id});

              }
          });
      });

      return results;
  }

  // Initialize the app
  init();
});
```

## 4.2 Test Cases:

### Test Case 1: **Play Music**

- **Description**: Test the functionality of selecting a file and playing it.
- **Steps**:
    1. Click the "Play" button.
    2. Select a valid `.mp3` or `.wav` file from the file dialog.
    3. Verify that the song starts playing.
    4. Verify that the song status label updates to "Playing: [song name]".
- **Expected Result**: The music should start playing, and the status label should show the song name.

### Test Case 2: **Pause Music**

- **Description**: Test the pause functionality when a song is playing.
- **Pre-condition**: A song is currently playing.
- **Steps**:
    1. Click the "Pause" button.
    2. Verify that the music pauses.
    3. Verify that the song status label updates to "Music Paused".
- **Expected Result**: The music should pause, and the status label should show "Music Paused".

## Test Case 3: **Unpause Music**

- **Description**: Test the unpause functionality when a song is paused.
- **Pre-condition**: A song is paused.
- **Steps**:
    1. Click the "Unpause" button.
    2. Verify that the music starts playing from the point it was paused.
    3. Verify that the song status label updates to "Music Playing".
- **Expected Result**: The music should resume from where it was paused, and the status label should show "Music Playing".

## Test Case 4: **Stop Music**

- **Description**: Test the stop functionality during playback.
- **Pre-condition**: A song is currently playing or paused.
- **Steps**:
    1. Click the "Stop" button.
    2. Verify that the music stops.
    3. Verify that the song status label updates to "Music Stopped".
- **Expected Result**: The music should stop playing, and the status label should show "Music Stopped".

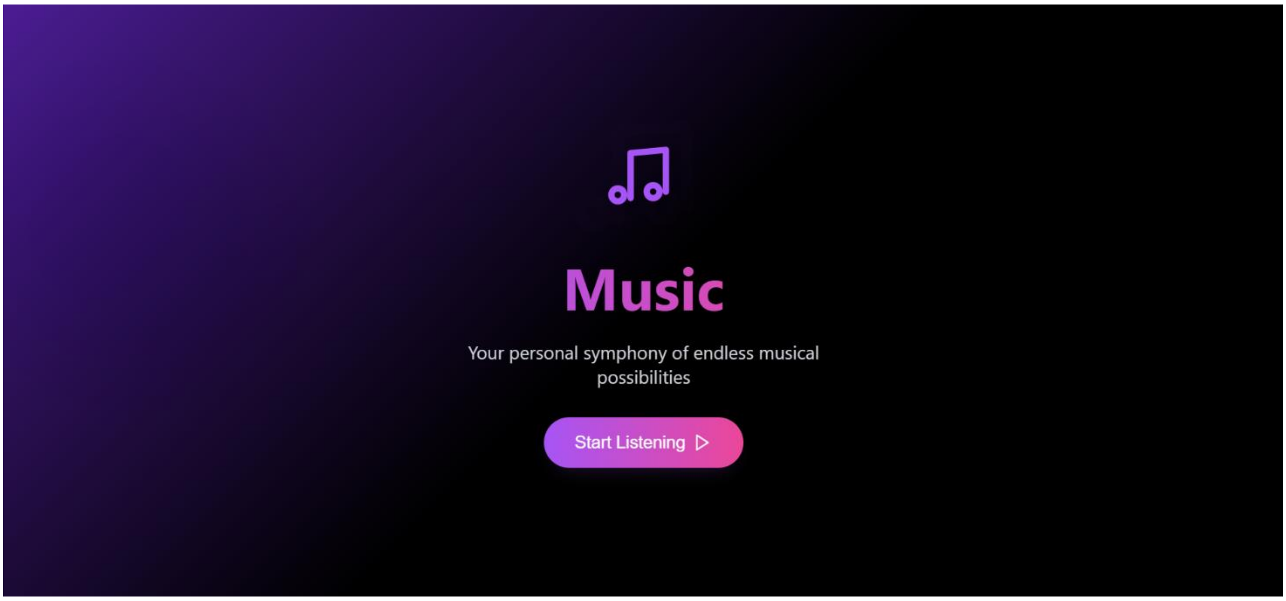
## Test Case 5: **Volume Control**

- **Description**: Test the volume slider functionality.
- **Pre-condition**: A song is playing.
- **Steps**:
    1. Move the volume slider to different values (e.g., 0%, 50%, 100%).
    2. Verify that the volume changes accordingly.
    3. Verify that the audio volume adjusts based on the slider value.
- **Expected Result**: The volume should change as per the slider value, and the music should be audible at the selected volume level.

# 5. RESULTS

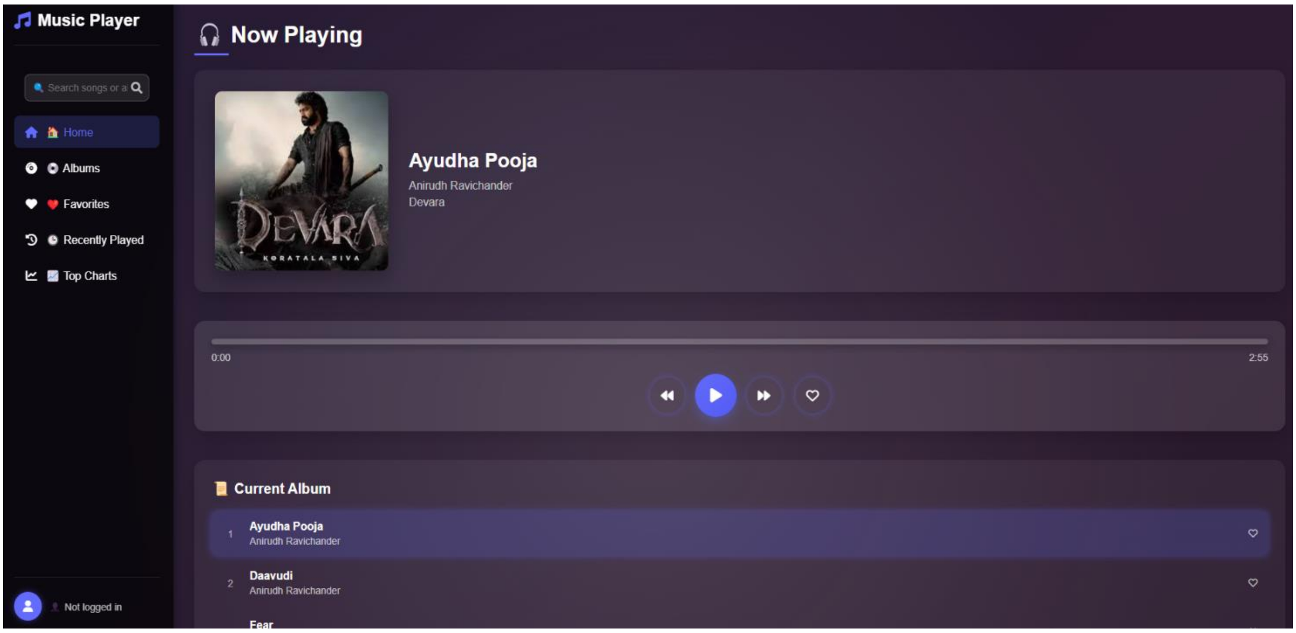
## 5.1 Output Screens:

### Homepage:

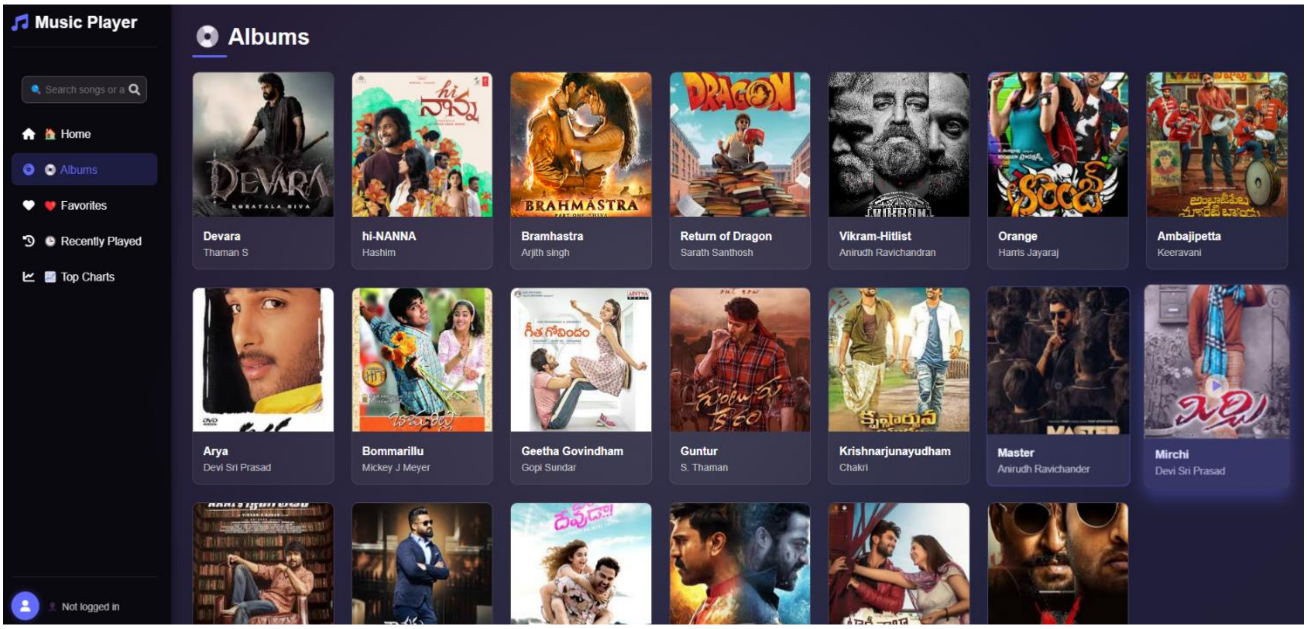


Your personal symphony of endless musical possibilities

### Music Player:



## Albums:



Albums

Devara
Thaman S

hi-NANNA
Hashim

Bramhastra
Arjith singh

Return of Dragon
Sarath Santhosh

Vikram-Hitlist
Anirudh Ravichandran

Orange
Harris Jayaraj

Ambajipetta
Keeravani

Arya
Devi Sri Prasad

Bommarillu
Mickey J Meyer

Geetha Govindham
Gopi Sundar

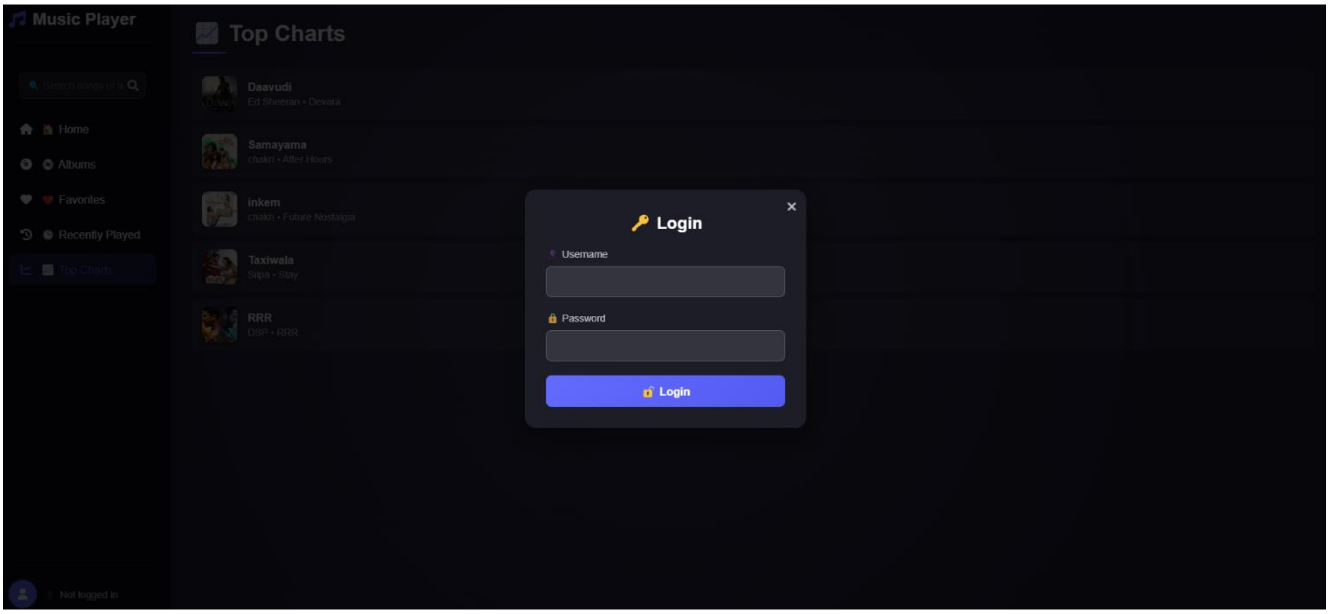
Guntur
S. Thaman

Krishnarjunayudham
Chakri

Master
Anirudh Ravichander

Mirchi
Devi Sri Prasad

## Login Logout:



Top Charts

Daavudi
Ed Sheeran • Devara

Samayama
chakri • After Hours

inkem
chakri • Future Nostalgia

Taxiwala
Silpa • Stay

RRR
DSP • RRR

🔑 Login

Username

Password

🔓 Login

# 6.CONCLUSION

In conclusion,developing a web-bsed music player offers a valuable oppurtunity to create an engaging audio experience for users.By utilizing HTML,CSS,and JavaScript,we can build a user-friendly and responsive interface that allows users to play,pause,adjust volume,and navigate through tracks seamlessy. The music player application provides a seamless user experience, allowing users to easily navigate through their music library, create playlists, and control music playback. The application also incorporates advanced features such as music recommendation, equalizer settings, and lyrics display. The project demonstrates the importance of considering user needs and preferences in the design and development of a music player application. By incorporating user-centered design principles and leveraging cutting-edge technologies, the project has created a music player that enhances the overall music listening experience.

# REFERENCES

**W3Schools Tutorials and References**
***https://www.w3schools.com/***
**CSS-Tricks (Responsive Design Patterns)**
***https://css-tricks.com/***
**LinkedIn Learning - Web Development Courses**
***https://www.linkedin.com/learning/***

**Spotify**
***https://open.spotify.com/***

**GIT HUB LINK:**
***https://github.com/SaiVignesh55/MUSICPLAYER.git***