**CHAPTER 01**

**INTRODUCTION**

* 1. **Introduction:**

In today's digital a, Music streaming has fundamentally transformed the landscape of the music industry, providing users with unprecedented access to an immense catalog of songs spanning diverse genres, eras, and cultures. This revolution began with the advent of platforms like Spotify, Apple Music, and Pandora, which offer both free and subscription-based models. Subscribers enjoy ad-free listening, offline downloads, and enhanced features like higher audio quality.

One of the key advantages of music streaming is its convenience. Users can access their favorite songs, anywhere, as long as they have an internet connection. This accessibility has significantly reduced the reliance on physical formats like CDs, making music consumption more portable and flexible.

Moreover, streaming services leverage sophisticated algorithms and machine learning techniques to analyze users listening habits and preferences. This data-driven approach enables them to curate personalized playlists, recommendations, and radio stations tailored to individual tastes. As a result, listeners can effortlessly discover new artists, songs, and genres they might not have encountered otherwise, enriching their musical journey.

Furthermore, the rise of music streaming has democratized the music industry, providing independent and emerging artists with a global platform to showcase their talent. Unlike traditional distribution models, where artists heavily relied on record labels for promotion and exposure, streaming platforms empower musicians to directly connect with their fan base and earn royalties based on the number of streams.

In addition to fostering artist-fan relationships, streaming services also facilitate community engagement through features like collaborative playlists and social sharing. Users can create and share playlists with friends, follow their favorite artists, and participate in discussions within the platform's ecosystem, fostering a sense of belonging and camaraderie among music enthusiasts.

Overall, music streaming has reshaped not only how we listen to music but also how artists distribute, promote, and monetize their work. As technology continues to evolve and consumer preferences evolve, the streaming landscape is poised to continue its evolution, shaping the future of music consumption and creation.

## Statement of Problem:

While music streaming websites have revolutionized the way people access and enjoy music, they also face several challenges:

1. **Content Availability and Licensing:** Music streaming platforms must negotiate licensing deals with record labels and artists to make their music available for streaming. Sometimes, disputes over licensing terms can lead to content being removed from platforms temporarily or permanently, which can frustrate users who rely on these platforms for access to their favorite music.
2. **User Privacy and Data Security:** Music streaming websites collect a vast amount of user data, including listening habits, preferences, and personal information. Ensuring the privacy and security of this data is crucial to maintaining user trust. However, there have been concerns raised about data breaches, and unauthorized access of user data by streaming services or third parties.
3. **Discovery and Algorithm Bias:** While algorithms are used to personalize recommendations for users based on their listening history, there is a risk of algorithmic bias. Users may be exposed to a limited range of content or genres, leading to a lack of diversity in their musical discovery. Moreover, there have been allegations that algorithms favor major labels and mainstream artists over independent, further exacerbating inequalities in the music industry.
4. **Platform Fragmentation:** With multiple streaming platforms available, users may face the challenge of fragmentation, having to subscribe to multiple services to access their favorite artists or exclusive content. This can lead to subscription fatigue and frustration, as users navigate between different platforms to find the music they want.
5. **Artist Compensation Transparency:** There is often a lack of transparency in how streaming platforms calculate and distribute royalties to artists. Artists may struggle to understand their earnings from streaming, leading to confusion and mistrust in the system. Improved transparency and communication regarding royalty structures could help address these concerns.

Addressing these challenges requires collaboration between streaming platforms, artists, record labels, regulators, and consumers to develop fair and sustainable solutions that benefit all stakeholders in the music ecosystem.

## Objectives:

The objectives of music streaming websites typically revolve around providing an engaging and convenient platform for users to discover, access, and enjoy music while also supporting artists and rights holders. Here are some common objectives:

1. **Accessible Music Catalog:** Offer a vast and diverse catalog of songs spanning various genres, languages, and eras to cater to the diverse tastes and preferences of users.
2. **Convenience and Accessibility:** Provide users with convenient access to music anytime, anywhere, across multiple devices, including smartphones, tablets, computers, and smart speakers.
3. **Personalized Recommendations:** Utilize algorithms and machine learning techniques to deliver personalized recommendations, playlists, and radio stations based on users' listening history, preferences, and behavior.
4. **Artist Promotion and Discovery:** Facilitate the discovery of new and emerging artists by promoting their music through featured playlists, recommendations, and editorial content, thereby supporting a thriving and diverse music ecosystem.
5. **Fair Compensation for Artists:** Ensure fair and transparent compensation for artists, songwriters, and rights holders through equitable royalty structures and revenue-sharing mechanisms.
6. **Community Engagement and Interaction:** Foster a sense of community among users by providing features for social sharing, collaborative playlists, user-generated content, and engagement with artists and fellow music enthusiasts.
7. **Quality Audio Experience:** Deliver high-quality audio streaming with options for different bitrates and formats to accommodate varying preferences and audiophile standards.
8. **Continuous Improvement and Innovation:** Strive for ongoing improvement and innovation in user experience, content discovery, audio quality, and platform features through feedback, research, and technological advancements.

By aligning with these objectives, music streaming websites aim to create a compelling and rewarding experience for users while supporting the growth and vitality of the music industry ecosystem.

## Applications:

The Music streaming websites have a wide range of applications across various contexts, catering to the diverse needs and preferences of users. Some common applications include:

**Music Enthusiasts:**

* Music enthusiasts can use the platform to discover new artists, albums, and genres, expanding their musical horizons and enhancing their listening experience.
* They can create personalized playlists, share their favorite music with friends, and engage in discussions about music, fostering a sense of community and connection among fellow enthusiasts.

**Independent Artists:**

* Independent artists can leverage the platform to showcase their music to a global audience, gaining exposure and building a fan base without the need for traditional record labels.
* The platform provides artists with valuable data and insights into listener demographics, preferences, and engagement metrics, helping them tailor their marketing and promotional strategies.

**Music Industry Professionals:**

* Music industry professionals, such as producers, promoters, and managers, can use the platform to discover emerging talent, scout for new signings, and identify trends in the music landscape.
* They can access analytics and performance metrics for artists and tracks, aiding in decision-making processes related to marketing campaigns, and talent development.

**Educational Institutions:**

* Educational institutions, including schools, colleges, and music academies, can incorporate the platform into their curriculum for music education and appreciation.
* Students can access a vast library of music spanning different genres and historical periods, supplementing real-world examples and practical applications.

**Event Organizers:**

* Event organizers, such as concert promoters, festival planners, and venue managers, can utilize the platform to promote upcoming events, sell tickets, and engage with attendees.
* They can create curated playlists featuring artists scheduled to perform at their events, generating excitement and anticipation among potential attendees.

**Advertisers and Brands:**

* Advertisers and brands can partner with the platform to reach a highly engaged and targeted audience of music enthusiasts through sponsored playlists, branded content, and audio ads.
* They can leverage user data and insights to tailor their messaging and ad placements for maximum impact and relevance.

**Fitness Enthusiasts:**

* Fitness enthusiasts can utilize the platform to create customized workout playlists tailored to their exercise routines and preferences, helping them stay motivated and energized during workouts.
* They can access curated workout playlists curated by fitness experts and instructors, providing them with a seamless music experience that complements their fitness goals.

**Social Impact Initiatives:**

* Social impact initiatives and nonprofit organizations can partner with the platform to raise awareness about social issues, promote positive messages, and drive community engagement through music.
* They can curate playlists focused on specific social causes, host virtual concerts or fundraising events, and collaborate with artists who share their values and mission.

By serving the diverse needs and interests of these additional stakeholders, the music streaming website fosters a dynamic and inclusive ecosystem that extends beyond entertainment to encompass wellness, education, social impact, creativity, and professional development.

## Limitations:

Music streaming websites, while immensely popular and convenient, also face several limitations and challenges:

**Internet Dependency:** Streaming music requires a stable internet connection, making it inaccessible in areas with poor or limited internet connectivity. Moreover, streaming consumes data, which can be costly for users with limited data plans or in regions where internet access is expensive.

**Audio Quality:** Despite advancements in streaming technology, the audio quality of streamed music may not match that of physical formats like CDs or high-resolution digital downloads. Compression techniques used to reduce file sizes can result in loss of audio fidelity, especially for audiophiles or discerning listeners.

**Content Availability:** Music licensing agreements vary by region, leading to discrepancies in the availability of certain songs or albums across different countries. Users may encounter frustration when trying to access music that is unavailable in their region due to licensing restrictions.

**Platform Exclusivity:** Exclusive deals between streaming platforms and artists or labels can lead to fragmented content across different platforms. Users may need to subscribe to multiple services to access all their favorite music, leading to subscription fatigue and increased costs.

**Artist Compensation:** Artists often receive minimal royalties per stream from streaming platforms, leading to concerns about fair compensation for their work. Independent and emerging artists, in particular, may struggle to earn a sustainable income from streaming, despite the platform's global reach.

**Discovery Algorithms:** While algorithms are used to personalize recommendations and playlists for users, there is a risk of algorithmic bias. Users may be exposed to a limited range of content or genres, leading to a homogenized music discovery experience and limiting exposure for niche or independent musicians.

**Privacy Concerns:** Music streaming platforms collect vast amounts of user data, including listening habits, preferences, and personal information. Concerns have been raised about data privacy and security, as well as the potential misuse of user data by streaming services or third parties.

**Technical Issues:** Despite efforts to ensure a seamless streaming experience, users may encounter technical issues such as buffering, playback errors. Despite these limitations, our platform remains committed to improving incident reporting and community safety through technological innovation and collaboration with stakeholders. We continuously strive to address these challenges and enhance the effectiveness and accessibility of our services.

**CHAPTER 02**

**METHODOLOGY**

* 1. **Research:**

The research methodology for the Music Streaming Website involves a structured approach to gather insights, analyze industry trends, and understand user preferences. Here's a breakdown of the research approach:

1. **Market Analysis:** Conduct a comprehensive analysis of the music streaming industry, including market size, key players, revenue models, and emerging trends. This involves studying industry reports, market research publications, and financial analyses to identify opportunities and challenges in the market.
2. **User Feedback:** Engage with potential users, including music enthusiasts, artists, and industry professionals, through surveys and interviews. Gather feedback on their experiences with existing music streaming platforms, preferences for features, and pain points. This qualitative data will provide valuable insights into user needs and expectations.
3. **Competitor Analysis:** Analyze the features, pricing strategies, and user experience of competing music streaming platforms. This involves studying competitor websites, mobile apps, and user reviews to identify strengths and weaknesses. The findings will inform the design and development of the Music Streaming Website, ensuring it offers unique value propositions and competitive advantages.

By combining market analysis, user feedback, and competitor analysis, the Music Streaming Website can gain valuable insights to inform its strategy and decision-making process.

## Platform Development:

In order to develop the Music Streaming Website, a systematic approach will be followed to design and implement key features and functionalities. Here's an outline of the platform development methodology:

1. **Requirement Gathering:** Define the requirements and specifications for the Music Streaming Website based on research findings and stakeholder input. This involves documenting user stories, and feature prioritization to guide the development process.
2. **Prototyping:** Create wireframes and mockups to visualize the user interface and user experience of the Music Streaming Website. This early feedback and validation from stakeholders, ensuring the final product meets user expectations.
3. **Agile Development:** Adopt an agile development methodology to iteratively build and test the Music Streaming Website in sprints. This involves breaking down the development process into manageable tasks, conducting regular sprint planning, and continuous integration and deployment. By following agile principles.
4. **Quality Assurance:** Conduct rigorous testing of the Music Streaming Website to identify and resolve bugs, usability issues, and performance bottlenecks. This includes functional testing, usability testing, compatibility testing across devices and browsers, and security testing to ensure a robust and reliable platform.

By following a structured approach to platform development, the Music Streaming Website can ensure a seamless user experience, robust functionality, and successful launch in the market.

* 1. **Requirements:**

# CHAPTER 03 DESIGN

Requirements gathering lays the groundwork for the design phase, focusing on identifying and documenting the needs, preferences, and expectations of stakeholders. The design of the Music Streaming Website is informed by comprehensive research, including user surveys, interviews, and industry analysis. Key requirements identified include:

* **User-Centric Interface:** The Music Streaming Website aims to provide a user-centric interface that prioritizes ease of use and accessibility. Requirements were specified for intuitive navigation, responsive design, and customizable preferences to enhance the user experience across different devices and screen sizes.
* **Seamless Music Discovery:** A primary requirement of the Music Streaming Website is to facilitate seamless music discovery, allowing users to explore, discover, and curate personalized playlists. This includes features such as advanced search functionality, personalized recommendations, and curated playlists based on user preferences and listening history.
* **High-Quality Audio Streaming:** Given the importance of audio quality in music streaming, requirements were defined for high-quality audio streaming, supporting various bitrates and formats to cater to different preferences and devices. The platform aims to deliver an immersive listening experience with minimal latency and buffering.
* **Social Interaction and Engagement:** Stakeholders expressed the need for social interaction and engagement features to foster community among users and artists. Requirements were outlined for social sharing, collaborative playlists, and user-generated content, enabling users to connect, share, and interact with each other and their favorite artists.
* **Licensing and Copyright Compliance:** The Music Streaming Website must comply with licensing agreements and copyright laws to ensure the legal distribution of music and fair compensation for artists and rights holders. Requirements were established for content licensing, rights management, and copyright protection mechanisms to uphold legal and ethical standards.

## System Design:

The system design phase focuses on translating requirements into a scalable and robust architecture that forms the foundation of the Music Streaming Website. Design decisions are guided by considerations such as scalability, performance, security, and reliability. Key components of the system design include:

* **Architecture:** The Music Streaming Website adopts a distributed architecture, comprising frontend, backend, and database layers. The frontend handles user interface components and client-side interactions, while the backend manages server-side logic, data processing, and integration with external services. The database layer stores and retrieves user data, music metadata, and system configurations.
* **User Authentication and Authorization:** A robust authentication and authorization system is implemented to ensure secure access to the platform. Users are required to authenticate themselves using credentials such as email and password, with additional measures such as OAuth for third-party authentication. Role-based access control mechanisms are employed to manage user permissions and restrict access to sensitive features and data.
* **Scalability and Performance:** The Music Streaming Website is designed to be scalable and performant, capable of handling a large volume of concurrent users and music streams. Techniques such as load balancing, caching, and horizontal scaling are employed to optimize performance and ensure responsiveness, even during peak usage periods.
* **Data Management:** The system design includes provisions for efficient data management, including data storage, retrieval, and manipulation. A distributed database management system (DBMS) such as PostgreSQL or MongoDB is utilized for structured and unstructured data storage, with mechanisms for data replication, sharding, and backup to ensure data reliability and availability.
* **Content Delivery Network (CDN):** To optimize content delivery and minimize latency, a CDN is employed to cache and distribute music files and multimedia content across geographically distributed servers. This helps improve streaming performance and reduce bandwidth usage, resulting in faster load times and smoother playback for users worldwide.

## Technologies Used:

The selection of appropriate technologies plays a critical role in the design and development of Safe Society, influencing factors such as functionality, performance, scalability, and security. The technologies chosen for Safe Society include:

* + - **Frontend Technologies**: HTML, CSS, and JavaScript are used for frontend development, enabling the creation of responsive and visually appealing user interfaces. Frameworks such as Bootstrap and Vue.js may be employed to streamline development and enhance interactivity.
    - **Backend Technologies**: PHP is chosen as the primary backend programming language for its versatility, performance, and extensive ecosystem of libraries and frameworks. Frameworks such as Laravel or CodeIgniter may be utilized to facilitate rapid development and maintainability.
    - **Database Management System**: MySQL is selected as the database management system (DBMS) for its reliability, scalability, and robust feature set. MySQL provides support for ACID transactions, data integrity constraints, and efficient indexing, making it suitable for handling structured data in Safe Society.

**XAMPP**: XAMPP is an open-source, cross-platform web server solution that facilitates the development and testing of web applications locally on a personal computer. The acronym "XAMPP" stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P). It includes all the necessary components required to set up a local web server environment, allowing developers to work on web projects without the need for a remote server.

* + - **Apache**: Apache is a widely-used, open-source web server software that powers a significant portion of websites on the internet. In XAMPP, Apache serves as the HTTP server component, responsible for processing and delivering web content to client browsers. It supports various features such as virtual hosting, SSL/TLS encryption, URL rewriting, and server-side scripting.
    - **MySQL**: MySQL is an open-source relational database management system (RDBMS) that is widely used for storing and managing structured data. In XAMPP, MySQL is included as the database server component, allowing developers to create and interact with MySQL databases locally. It provides support for SQL queries, data manipulation, transactions, and database administration tasks.

**phpMyAdmin**: phpMyAdmin is a free and open-source web-based application written in PHP that provides a graphical interface for managing MySQL databases. It allows users to perform various database administration tasks, such as creating, modifying, and deleting databases, tables, and columns; executing SQL queries; importing and exporting data; managing user permissions; and monitoring database performance.

* 1. **Front-end: Home Page:**

# CHAPTER 04 IMPLEMENTATION

* + - Design a visually appealing home page layout that captures the essence of Music Streaming Website.
    - Include prominent sections for key elements such as songs, master player, about, search, and login forms.
    - Use high-quality images, songs, and icons to enhance the visual appeal of the home page.
    - Implement responsive design principles to ensure compatibility with various devices and screen sizes.

## Master Player:

* + - Develop a Master Music Player for playing songs or playing a specific song and also Next, Previous and Pause buttons.
    - Next button helps to play the next song.
    - Previous button helps to play the previous song.
    - Pause button helps to pause the song or to play the paused song.
    - There is also a progress bar which shows the progress of song played and remaining time of song.
    - There is also a wave element which animates while playing the song and volume bar using which you can also control volume of song.

## Backend:

1. **Choose Backend Technology:**
   * Select a suitable backend technology stack based on project requirements and developer expertise. This could be PHP.

## Setup Development Environment:

* + Install necessary software tools such as a text editor or IDE, version control system (e.g., Git), and local development server (e.g., XAMPP, WAMP, or MAMP).
  + Install and configure the chosen backend framework and dependencies.

## Backend Logic:

* + Develop backend logic to handle business processes, user authentication, authorization, and API endpoints.
  + Implement user authentication mechanisms such as JWT, OAuth2, or session- based authentication.
  + Write code to handle CRUD operations for interacting with the database.
  + Implement validation and error handling to ensure data integrity and provide meaningful feedback to users.

## Database:

1. **Database Selection:**
   * Choose an appropriate database management system (DBMS) based on project requirements, scalability needs, and data modeling preferences. Options include MySQL, PostgreSQL, MongoDB, etc.

## Database Schema Design:

* + Design the database schema to model the application's data structure effectively.
  + Identify entities, attributes, and relationships and define them in the database schema.
  + Choose appropriate data types, constraints, and indexes to ensure data integrity and performance.

## Table Creation:

* + Create database tables based on the defined schema using SQL or ORM tools provided by the backend framework.
  + Define primary keys, foreign keys, unique constraints, and indexes to enforce data integrity and optimize query performance.

## Database Administration:

* + Assign roles and permissions to database users to control access to data and database management functions.
  + Monitor database performance, usage metrics, and resource utilization to identify potential issues and optimize database performance.
  + Perform routine maintenance tasks such as index rebuilding, statistics updating, and database optimization to ensure optimal performance and reliability.

## 5.1 Code:

**Home page:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Music Website</title>

<link

rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"

/>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<header>

<div class="menu\_side">

<h1>Playlist</h1>

<div class="playlist">

<h4 class="active">

<span></span><i class="bi bi-music-note-beamed"></i> Playlist

</h4>

<br />

<!-- <h4>

<span></span><i class="bi bi-music-note-beamed"></i>Last Listend

</h4> -->

<h4>

<span></span><i class="bi bi-music-note-beamed"></i>Recommended

</h4>

</div>

<div class="menu\_song">

<li class="songItem">

<span>01</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="1"></i>

</li>

<li class="songItem">

<span>02</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="2"></i>

</li>

<li class="songItem">

<span>03</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="3"></i>

</li>

<li class="songItem">

<span>04</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="4"></i>

</li>

<li class="songItem">

<span>05</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="5"></i>

</li>

<li class="songItem">

<span>06</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="6"></i>

</li>

<li class="songItem">

<span>07</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="7"></i>

</li>

<li class="songItem">

<span>08</span>

<img src="img/01.webp" alt="" />

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

<div class="content">

<h1>Alen Walker-Faded</h1>

<p>

You were the shadow to my life did you feel us Another start you

<br />Away afraid our aim is out of sight Wanna see us alive

</p>

<div class="buttons">

<button class="play">PLAY</button>

<button class="follow">FOLLOW</button>

</div>

</div>

<div class="popular\_song">

<div class="h4">

<h4>Popular Song</h4>

<div class="btn\_s">

<i class="bi bi-arrow-left-short" id="pop\_song\_left"></i>

<i class="bi bi-arrow-right-short" id="pop\_song\_right"></i>

</div>

</div>

<div class="pop\_song">

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="9"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="10"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="11"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="12"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="13"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="14"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="15"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="16"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="17"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="18"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="19"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="20"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="21"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

</h5>

<i class="bi playlistplay bi-play-circle-fill" id="8"></i>

</li>

</div>

</div>

<div class="song\_side">

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="22"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

<li class="songItem">

<div class="img\_play">

<img src="img/01.webp" alt="" />

<i class="bi playlistplay bi-play-circle-fill" id="23"></i>

</div>

<h5>

On My Way <br />

<div class="subtitle">Alan Walker</div>

</h5>

</li>

</div>

</div>

<div class="popular\_artist">

<div class="h4">

<h4>Popular Artists</h4>

<div class="btn\_s">

<i class="bi bi-arrow-left-short" id="pop\_art\_left"></i>

<i class="bi bi-arrow-right-short" id="pop\_art\_right"></i>

</div>

</div>

<div class="item">

<li>

<a href="arjith.php"> <img src="img/dp.webp" alt="" /></a>

</li>

<li>

<a href="keeravani.php"> <img src="img/dp2.webp" alt="" /></a>

</li>

<!-- <li>

<img src="img/dp2.webp" alt="keeravani.php" />

</li>

<li>

<img src="img/dp2.webp" alt="keeravani.php" />

</li>

<li>

<img src="img/dp2.webp" alt="keeravani.php" />

</li> -->

</div>

</div>

</div>

<div class="master\_play">

<div class="wave" id="wave">

<div class="wave1"></div>

<div class="wave1"></div>

<div class="wave1"></div>

</div>

<img id="poster\_master\_play" src="img/1.webp" alt="" />

<h5 id="title">

Vande matharam

<div class="subtitle">Thaman Bro</div>

</h5>

<div class="icon">

<i class="bi shuffle bi-music-note-beamed">next</i>

<i class="bi bi-skip-start-fill" id="back"></i>

<i class="bi bi-play-fill" id="masterPlay"></i>

<i class="bi bi-skip-end-fill" id="next"></i>

<a href="" download id="download\_music">

<i class="bi bi-cloud-arrow-down-fill"></i

></a>

</div>

<div class="icon">

<i class="bi shuffle bi-music-note-beamed">next</i>

<i class="bi bi-skip-start-fill" id="back"></i>

<i class="bi bi-play-fill" id="masterPlay"></i>

<i class="bi bi-skip-end-fill" id="next"></i>

<a href="" download id="download\_music">

<i class="bi bi-cloud-arrow-down-fill"></i

></a>

</div>

<span id="currentStart">0:00</span>

<div class="bar">

<input type="range" id="seek" min="0" max="100" />

<div class="bar2" id="bar2"></div>

<div class="dot"></div>

</div>

<span id="currentEnd">0:30</span>

<div class="vol">

<i class="bi bi-volume-up-fill" id="vol\_icon"></i>

<input type="range" min="0" max="100" id="vol" />

<div class="vol\_bar"></div>

<div class="dot" id="vol\_dot"></div>

</div>

</div>

</header>

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

</body>

</html>

## Signup PHP:

<?php

session\_start();

include('db\_connect.php');

$msg = false;

if ($\_SERVER['REQUEST\_METHOD']=='POST') {

$user\_name = $\_POST['user\_name'];

$user\_email = $\_POST['user\_email'];

$user\_password = $\_POST['user\_password'];

$user\_re\_password = $\_POST['user\_re\_password'];

if (!empty($user\_name) && !empty($user\_email) && !empty($user\_password) && !is\_numeric($user\_name)) {

if($user\_password === $user\_re\_password ){

$query = "insert into user (user, email, password ) VALUES ('$user\_name','$user\_email','$user\_password')";

mysqli\_query($con, $query);

header("Location: login.php");

}

else{

$msg = "Password Not Match";

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Music Website Sign Up</title>

<link

rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"

/>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<header>

<div class="left\_bx1">

<div class="content">

<form method="post">

<h3>Sign Up</h3>

<div class="card">

<label for="name">Name</label>

<input

type="text"

name="user\_name"

placeholder="Enter Your Username..."

required

/>

</div>

<div class="card">

<label for="email">Email</label>

<input

type="email"

name="user\_email"

placeholder="Enter Your email..."

required

/>

</div>

<div class="card">

<label for="password">Password</label>

<input

type="password"

name="user\_password"

placeholder="Enter Your Password..."

required

/>

</div>

<div class="card">

<label for="re-password">Re-Password</label>

<input

type="password"

name="user\_re\_password"

placeholder="Enter Your Re-Password..."

required

/>

</div>

<input type="submit" value="Sign Up" class="submit" />

<div class="check">

<input type="checkbox" name="" id="" /><span>Remember Me.</span>

</div>

<p>You Have a account?<a href="login.html">Login</a></p>

</form>

</div>

</div>

<div class="right\_bx1">

<img src="img/musiclogo.png" height="400px" width="400px" alt="" />

<!-- <h3>Incorrect Password</h3> -->

<?php

echo( '<h3>'.$msg.'</h3>');

?>

</div>

</header>

</body>

</html>

<?php session\_start();

require\_once 'db\_connection.php';

$name = $\_POST['name'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$select = "SELECT \* FROM user where email='$email'";

$getaUser = mysqli\_query($conn, $select); if(mysqli\_num\_rows($getaUser) == 1){

$\_SESSION['message'] = "error";

$\_SESSION['error'] = "Already user exists"; unset($\_SESSION['success']); header("Location: signup.php");

exit();

} else {

$photo = './default.png'; // Adjust the path as necessary

$defaultImage = './images/default.png'; // Adjust the path as necessary

$imageContent = file\_get\_contents($defaultImage);

$imageContent = base64\_encode($imageContent);

//$encPass = password\_hash($password, PASSWORD\_BCRYPT);

$insert = "INSERT INTO user (name, email, password, photo) values ('$name', '$email', '$password', '$imageContent')";

mysqli\_query($conn, $insert);

$\_SESSION['message'] = "success";

$\_SESSION['success'] = "Successfully registered"; unset($\_SESSION['error']);

header("Location: signup.php"); exit();

}

?>

## Login PHP:

<?php

session\_start();

include('db\_connect.php');

$msg = false;

if (isset($\_POST['user\_name'])) {

$user\_name = $\_POST['user\_name'];

$user\_password = $\_POST['user\_password'];

$query = "select \* from user where user = '".$user\_name."' AND password = '".$user\_password."' limit 1";

$result = mysqli\_query($con,$query);

if (mysqli\_num\_rows($result)==1) {

header('Location: welcome.php');

} else {

$msg = "Incorrect Password";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Music Website Login</title>

<link

rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"

/>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<header>

<div class="left\_bx1">

<div class="content">

<form method="post">

<h3>Login</h3>

<div class="card">

<label for="name">Name</label>

<input

type="text"

name="user\_name"

placeholder="Enter Your Username..."

required

/>

</div>

<div class="card">

<label for="password">Password</label>

<input

type="password"

name="user\_password"

placeholder="Enter Your Password..."

required

/>

</div>

<input type="submit" value="Login" class="submit" />

<div class="check">

<input type="checkbox" name="" id="" /><span>Remember Me.</span>

</div>

<p>Don't have a account yet?<a href="signup.html">Sing Up</a></p>

</form>

</div>

</div>

<div class="right\_bx1">

<img src="" alt="" />

<!-- <h3>Incorrect Password</h3> -->

<?php

echo( '<h3>'.$msg.'</h3>');

?>

</div>

</header>

</body>

</html>

if ($user['role'] == 'admin') {

$\_SESSION['admin\_session\_token'] = bin2hex(random\_bytes(32)); // Generates a random 32-byte session token for admin

// Redirect to admin dashboard after successful login header("Location: admin.php");

exit();

} else if ($user['role'] == 'user'){

$\_SESSION['user\_session\_token'] = bin2hex(random\_bytes(32)); // Generates a random 32-byte session token for user

// Redirect to user dashboard after successful login header("Location: user.php");

exit();

} else if($user['role'] == 'head'){

$\_SESSION['head\_session\_token'] = bin2hex(random\_bytes(32)); // Generates a random 32-byte session token for head

// Redirect to head dashboard after successful login header("Location: head.php");

exit();

}

} else {

$\_SESSION['message'] = "error";

$\_SESSION['error'] = "Invalid password"; unset($\_SESSION['success']); header("Location: signin.php");

exit();

}

} else {

$\_SESSION['message'] = "error";

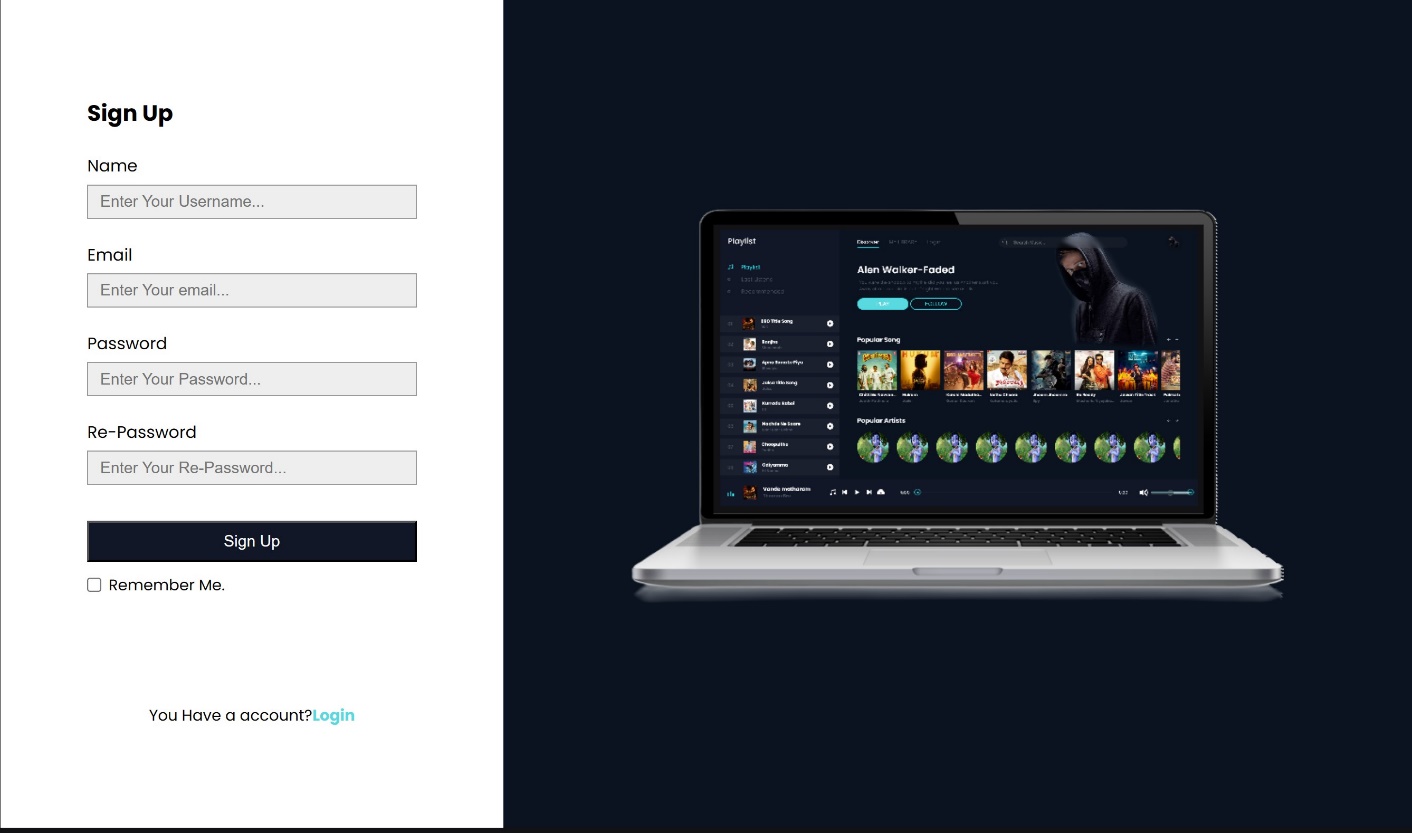
$\_SESSION['error'] = "Invalid email"; unset($\_SESSION['success']); header("Location: signin.php");

exit();

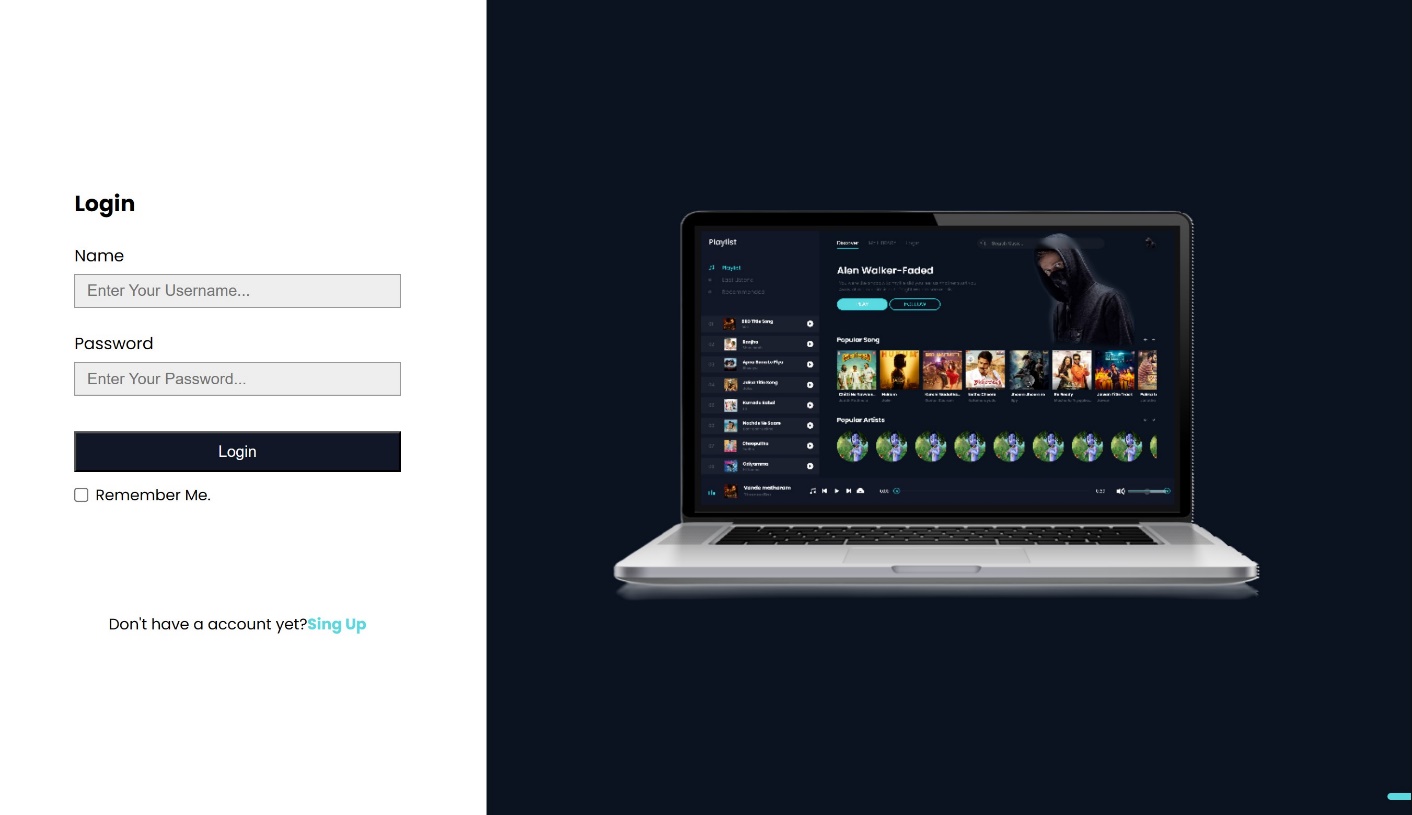
}

## 5.2 Result: Home Page:

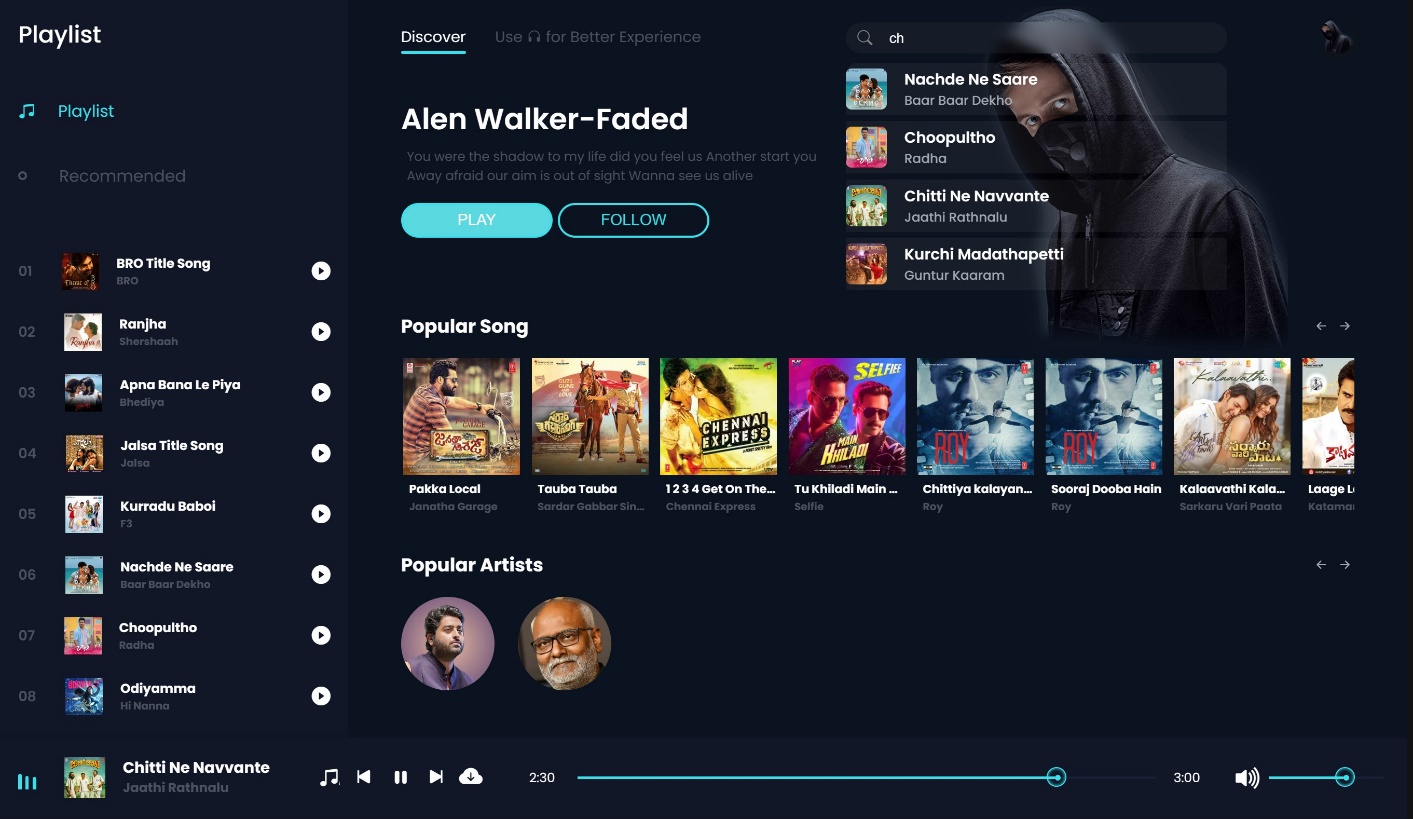
**Sign up Page:**

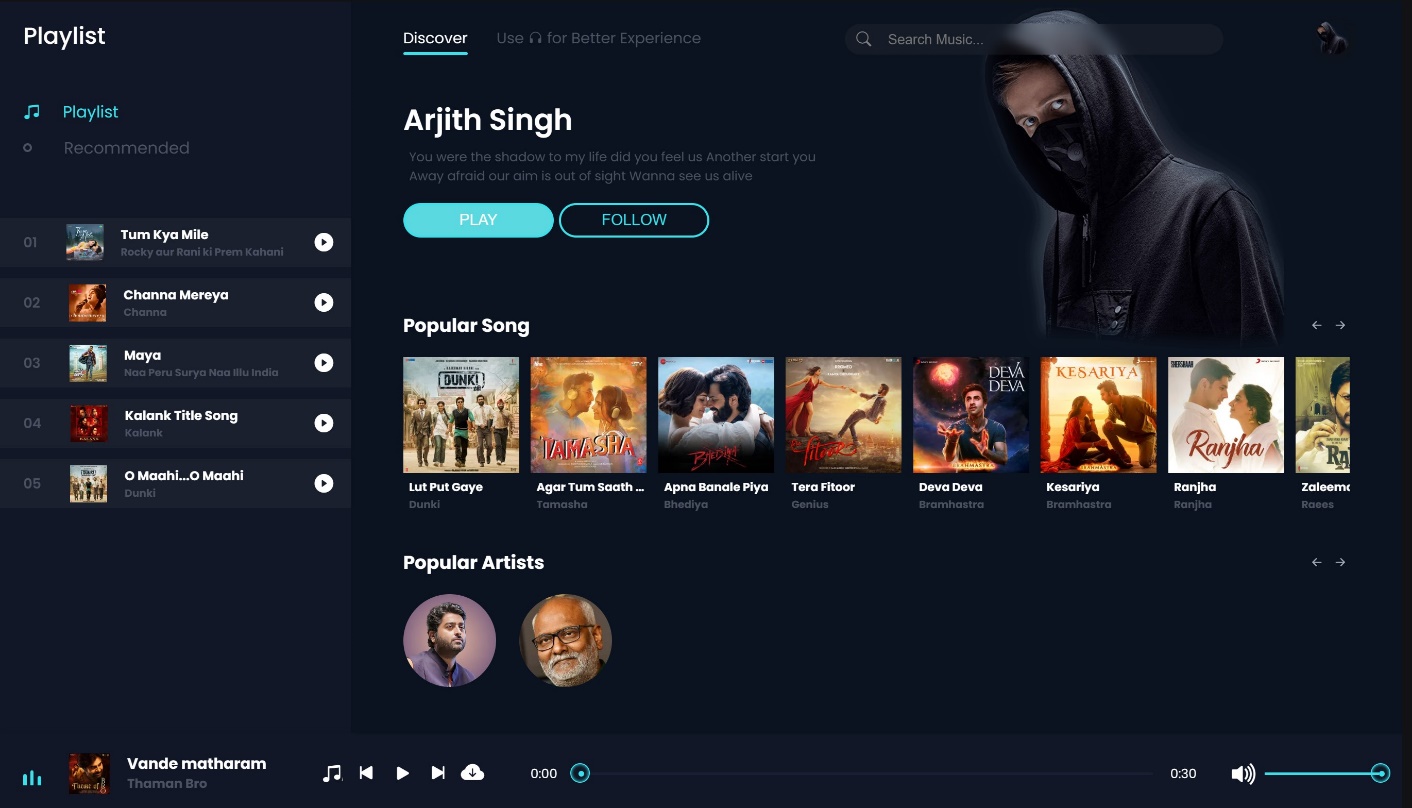


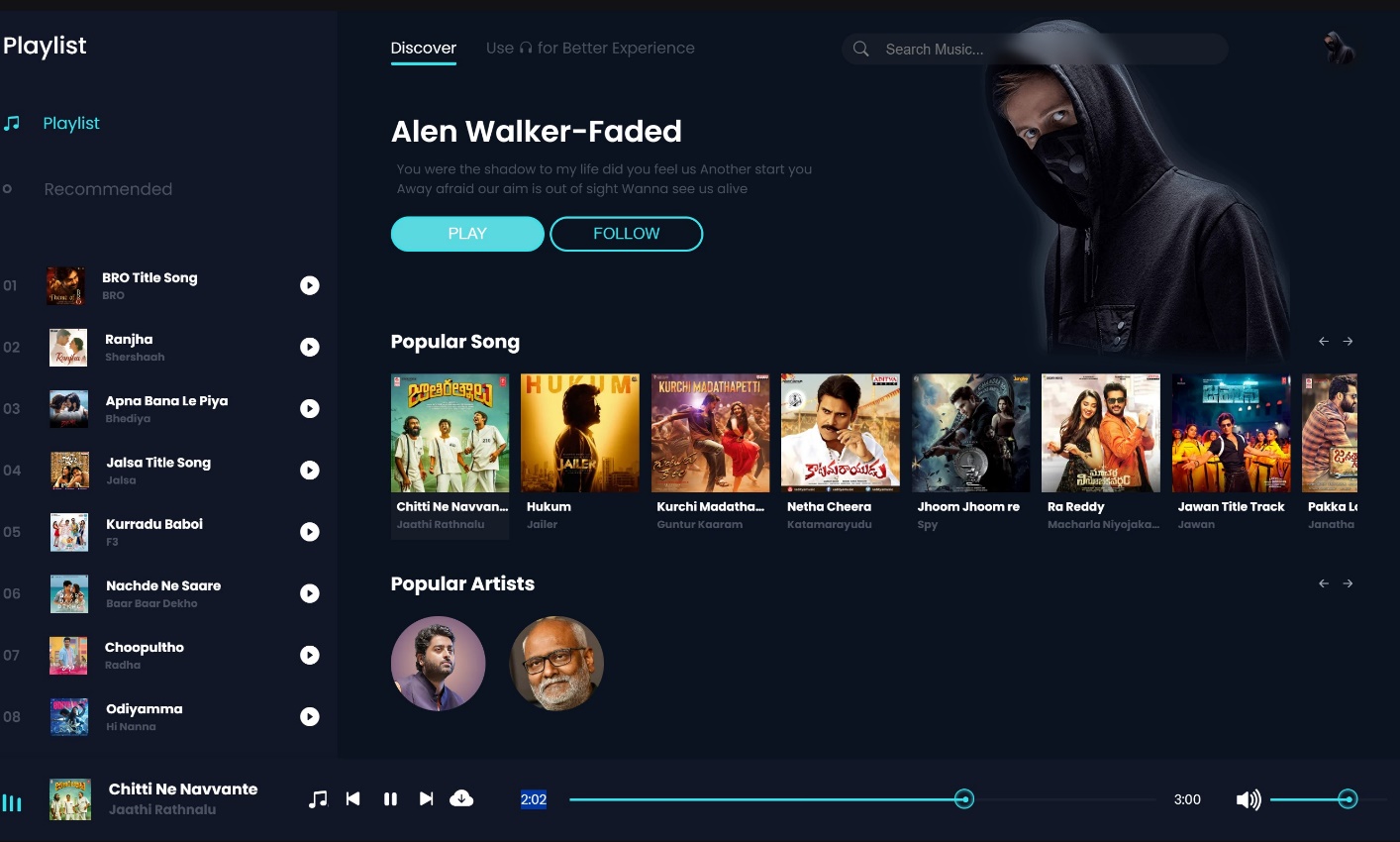
## Log in Page:

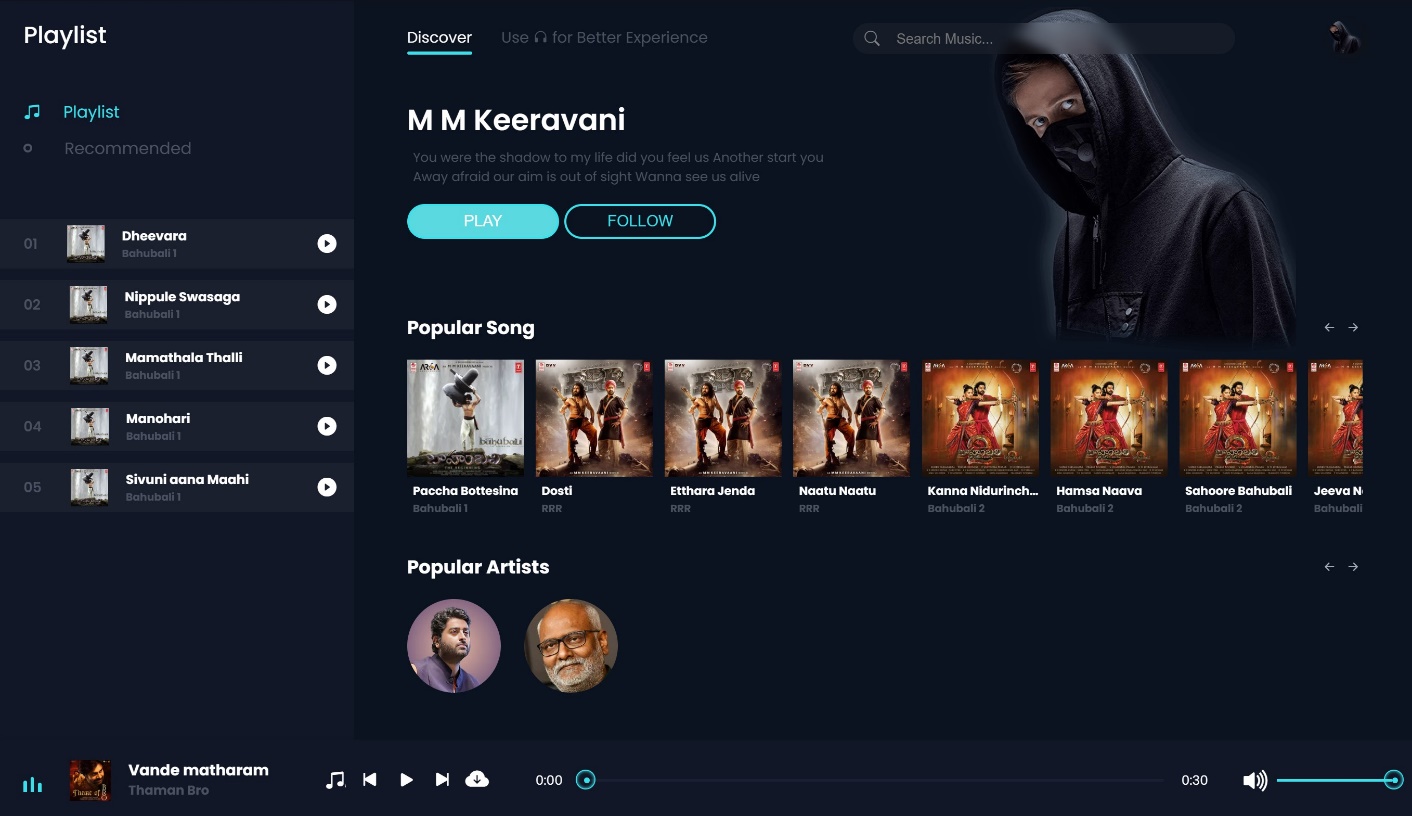


**Main Page:**









**6.1 Conclusion:**

# CHAPTER 06 CONCLUSION AND FUTURE WORK

In conclusion, the development of the music streaming website using HTML, CSS, JavaScript, and XAMPP has been a fulfilling journey that has not only met but also exceeded the initial project objectives. Through meticulous planning, persistent effort, and collaborative teamwork, we have successfully created a platform that offers an immersive and seamless music streaming experience.

Throughout the development process, we encountered several challenges, including optimizing database queries for efficient data retrieval, ensuring cross-browser compatibility for a consistent user experience, and implementing secure user authentication mechanisms to protect user data. However, through diligent problem-solving and leveraging the knowledge gained from our coursework and research, we were able to overcome these obstacles and deliver a robust and reliable solution.

## Moreover, this project has provided invaluable learning opportunities, allowing us to deepen our understanding of web development principles, database management techniques, and frontend design practices. We have gained practical experience in coding, debugging, and testing, honing our skills as aspiring developers and engineers.

## Looking ahead, there are several avenues for future improvement and expansion. This includes refining existing functionalities, such as enhancing the recommendation algorithm to provide more personalized suggestions and optimizing the performance of the website to handle larger user traffic. Additionally, integrating new features such as user-generated content, real-time chat support, and integration with external APIs could further enrich the platform and differentiate it from competitors.

## 6.2 Future Work:

While the current version of the music streaming website represents a significant achievement, there are several areas where further development and refinement could enhance the user experience and functionality. Future work on the project could include:

* Enhanced Recommendation Engine: Implementing a more sophisticated recommendation algorithm that takes into account user preferences, listening history, and social interactions to provide highly personalized music recommendations. This could involve machine learning techniques such as collaborative filtering or content-based filtering.
* Social Features: Introducing social networking features that allow users to connect with friends, share playlists, and discover new music based on the recommendations of their peers. This could include features such as user profiles, follower/following relationships, and news feeds displaying recent activity.
* User-Generated Content: Allowing users to upload and share their own music tracks, playlists, and podcasts on the platform. Implementing features such as user-generated playlists, comments, and ratings could foster community engagement and diversity of content.
* Integration with External APIs: Integrating with external APIs from music streaming services, social media platforms, or music metadata providers to expand the range of available content and enhance the accuracy of music recommendations. This could also facilitate features such as importing playlists from other services or sharing activity with social media networks.
* Mobile Application Development: Developing native mobile applications for iOS and Android platforms to provide a more seamless and optimized user experience on mobile devices. This could involve redesigning the user interface to accommodate smaller screens and implementing features such as offline playback and push notifications.
* Performance Optimization: Conducting performance optimization to improve the speed, scalability, and responsiveness of the website, particularly during peak usage periods. This could involve optimizing database queries, caching frequently accessed data, and implementing server-side caching techniques.
* Accessibility Enhancements: Ensuring that the website is accessible to users with disabilities by implementing accessibility features such as keyboard navigation, screen reader compatibility, and alternative text for multimedia content. This would help ensure that all users can enjoy the music streaming experience regardless of their abilities.

By focusing on these areas of future work, the music streaming website can continue to evolve and innovate, providing an even more compelling and enjoyable experience for users while remaining competitive in the ever-changing landscape of online music consumption.