PROJECT REPORT

IN

“COMPUTER SCIENCE ENGINEERING”

ON

“RESOUND MUSIC PLAYER”

SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE

OF

BE (CSE)

Under the Guidance of:
Name: Dr. Abhishek Thakur
Designation: Mentor
Department: Computer Science Engineering

Submitted By:
Madhav Goyal (2011981262)
Akhilesh Thakur (2111981020)
Akshita (2111981026)
Anjori (2111981033)

Chitkara University, Himachal Pradesh, India

CONTENTS

DECLARATION

We hereby declare that the project work titled, "**Resound | An Interactive Music Sharing and Streaming Platform**" submitted as part of Bachelor's degree in CSE, at Chitkara University, Himachal Pradesh, is an authentic record of our own work carried out under the supervision of Dr. Abhishek Thakur.

Signature(s):

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to our project supervisor, Dr. Abhishek Thakur, for his guidance, support, and expertise throughout the development of our music player. We are grateful for his invaluable insights and feedback that helped us refine our ideas and make critical decisions.

We would also like to thank our team members, Madhav Goyal, Akhilesh Thakur, Akshita and Anjori, for their hard work and dedication in bringing this project to fruition. Each team member brought unique skills and perspectives to the project, and their collaboration was essential to the success of the music player.

We extend our gratitude to the outside consultants who generously shared their expertise with us. Their insights and feedback helped us refine the functionality and usability of our music player. We would like to express our gratitude towards our parents member of CSE Department for their kind co-operation and encouragement which helps us in the completion of the project.

Lastly, we thank all of the users who provided valuable feedback that helped us improve the music player. We are proud of what we have accomplished and we hope that our music player will bring joy and entertainment to music lovers everywhere.

Thank you all for being a part of this journey and for making this music player project a success.

LIST OF FIGURES AND TABLES

PROJECT TITLE

Resound | An Interactive Music Sharing and Streaming Platform

INTRODUCTION

Welcome to the world of Music! Music is an integral part of our lives and has the power to evoke emotions, memories, and moods. Whether you're an avid music lover or just enjoy listening to your favourite tracks, our music player has everything you need to enhance your listening experience. In today's digital age, music players have become an essential tool for people to listen to their favourite music anytime, anywhere. The objective of this project is to design and develop a music player that offers the user with an easy-to-use interface. So, sit back, relax, and immerse yourself in the world of music with our music player.

In this report, we will discuss the development of our music player, including the design, functionality, and features. We will also highlight the challenges we faced and the solutions we implemented. Our goal is to provide a comprehensive overview of our project and demonstrate how our music player can meet the needs of today's music lovers.

Project Category:

In my view this project would fall under the category of web development, with a focus on front-end development. The front-end development aspect of the project would involve designing and developing the user interface of the website, including creating a navigation menu, designing the music player interface.

For the music playback functionality, the project could involve using an open-source music player library or creating a custom music player using HTML5 audio tag and JavaScript. The website could also include features such as searching for songs and displaying song lyrics. In this project, we need to design the user interface and user experience of the application, including features such as playlists, music library management, and playback controls.

ABSTRACT

The aim of this project is to develop a music player website that provides users with a comprehensive and user-friendly platform to listen to their favourite music. The website will be designed to allow users to access a vast collection of songs. This project aims to design and develop a music player that offers a user-friendly interface. Our approach involves integrating cutting-edge audio technology with intuitive design principles to create a music player that meets the needs of music lovers.

The project provides seamless integration with the music player, allowing users to control their music playback and access their music library on the go. Our project has the potential to improve the music listening experience for people around the world and set a new standard for music player design and functionality. To enhance the user experience, the website will include a search function that enables users to search for songs based on artists, genres, and keywords. The website will also feature a song recommendation system that suggests new songs based on the user's listening history.

The outcome of the project will be a functional music player website that provides an enjoyable and immersive music listening experience for users. The website will be a valuable resource for music lovers, allowing them to discover new music.

WORK DONE

Overview:

User Interface Design: This involves designing an intuitive and user-friendly interface for the music player that allows users to easily browse, search, and play their music.

Audio Playback Functionality: This involves implementing music playback controls, volume adjustments, and playlist management using an open-source music player library or custom music player

Music Library Management: This involves managing the user's music library, including adding/removing music, organizing music by artist/album/genre.

Audio Equalization: This involves adding equalization controls to the music player, which allows users to adjust the frequency response of the audio output to suit their preferences.

Front-end Development: This involves designing and developing the user interface and user experience of the website using HTML, CSS, and JavaScript. For a music player website, front-end development includes designing the music player interface, playlist management, and search functionality.

Testing and Optimization: This involves testing the website's functionality, performance, and security to ensure a seamless user experience. It also includes optimizing the website's design and functionality.

Purpose:

l. To allow users to play and listen to music.
2. Allow users to listen to music whenever and wherever they want.
3. To customize their listening experience to suit their preferences.
4. Easy access to a wide variety of music: A music player website provides users with easy access to a vast collection of songs, including popular music and niche genres.
5. Gives Convenience to the user.
6. User-friendly platform to listen to their favourite music and discover new songs.

Overall Description:

1. Product Perspective:

This software system is user friendly. The product has various features, such as the ability to manage a music library and adjust audio settings.

Some key aspects of the product perspective include:

Market analysis: Understanding the market for online music streaming services, including market size, competition, and user demographics.

Product features: Identifying the key features that make the music player website unique, such as personalized playlists, social sharing, and music recommendation algorithms.

User experience: Focusing on creating a seamless and enjoyable user experience that is easy to navigate and use.

Scalability: Ensuring that the website is scalable to handle increasing user traffic and a growing music library.

Business model: Identifying the revenue streams for the music player website, such as advertising, subscription fees, or a combination of both.

2. Product Function:

The product function of a music player is to allow users to play and listen to music. The core function of a music player is to read audio files, decode them, and play them.

Music library: The website should provide access to a vast collection of songs that users can search and browse.

Playlists: Users should be able to create customized playlists with their favourite songs and organize them in a way that makes sense to them.

Playback controls: The website should provide playback controls such as play, pause, skip, and volume control.

User accounts: Users should be able to create accounts to save their preferences, playlists, and listening history.

3. **Operating Environment:**

   This project works on the following:

   1. Operating System: Windows 7 and higher.
   2. Text-Editor: VS Code.
   3. Technologies used: HTML, CSS, JavaScript..

4. **Design and Implementation Constraints:**

   1. Platform Compatibility
   2. Audio Format Support
   3. Resource Limitations
   4. Licensing and copyright
   5. Storage Capacity

External Interface Requirements:

1. **User Interface:**

   - It will be easy to use and user friendly.
   - It has standard playback controls.
   - User-friendly platform to listen to their favourite music and discover new songs.
   - To customize their listening experience to suit their preferences.
   - Easy access to a wide variety of music: A music player website provides users with easy access to a vast collection of songs, including popular music and niche genres.

2. **Software Interface:**

   This whole project works on browsers like Chrome, Firefox etc. and is based on the technologies like HTML, CSS and JavaScript. It has the following features:

   - Navigation: The website should have a clear and easy-to-use navigation system that allows users to access different functions and features.

- Search: The website should have a search bar that allows users to search for songs, artists, albums, and playlists.
- Playlists: Users should be able to manage playlists easily.
- Playback controls: The website should provide clear and accessible playback controls, such as play, pause, skip, and volume control.
- User accounts: Users should be able to create and manage their accounts easily, with the ability to view their listening history, manage their preferences.
- Design: The website should have an attractive and visually appealing design that is consistent across all pages and functions.

Other Non-Functional Requirements:

1. **Performance Requirements**
   - It gives quick responses.
   - It is time saving.
   - It is easily accessible to the user.
   - It has Log In/Log Out option.

2. **Portability Requirements**

   Cross-browser compatibility: The website should be designed to work seamlessly on different web browsers, including Chrome, Firefox, Safari, and Edge.

   User accounts: Users should be able to create accounts to save their preferences, playlists, and listening history, allowing them to access their music from any device.

3. **Availability Requirements**

   This music player is up and running whenever needed. The users can access the website at all times, without any significant interruptions or downtime. By meeting these requirements, a music player website can provide a reliable and trustworthy user experience, leading to increased engagement and user retention.

4. **Scalability Requirements**

Scalability requirements refer to the ability of a music player website to handle increasing user traffic, a growing music library, and new features. A music player website should be scalable to ensure that it can meet the demands of its users, without sacrificing performance or user experience. It meets the needs for which it was build.

5. **Security Requirements**

This system provides basic security authentication. Security requirements for a music player website are critical to ensure that user data is protected, and the website is not vulnerable to attacks or breaches. Some key security requirements for a music player website include:

- Authentication and authorization: The website should have secure user authentication and authorization mechanisms, ensuring that only authorized users can access the website and their data.
- Data encryption: The website should use encryption to protect user data, such as passwords, credit card information, and personal information.
- Secure communication: The website should use secure communication protocols, such as HTTPS, to protect user data in transit.
- Access controls: The website should have appropriate access controls to limit user access to sensitive data and functionality.

6. **Safety Requirements**

Safety requirements for a music player website typically focus on protecting users from harm while using the website. It keeps the records. Also, it is safe from various attacks.

User privacy: The website should have privacy controls and data protection measures in place to ensure that users are not at risk of identity theft, fraud, or other types of cybercrime.

7. **Usability Requirements**
    - Intuitive user interface: The website should have a clear and intuitive user interface that allows users to easily navigate the website and find the content they are looking for.
    - Simple controls: The website should have simple controls that are easy to understand and use, such as play/pause, skip, volume, and playlist controls.
    - Search functionality: The website should have a robust search function that allows users to easily find specific songs, albums, or artists.

8. **Reliability Requirements**

    Reliability requirements for a music player website focus on ensuring that the website operates reliably and consistently, without downtime or errors. Some key reliability requirements for a music player website include:
    - Uptime: The website should be available and accessible to users at all times, without extended periods of downtime. It is available to the user all the time.
    - Performance: The website should perform well and respond quickly, without lag or delays, to user interactions such as play/pause, skip, and playlist management.

9. **Supportability Requirements**

    It is supported by all the browsers like chrome, Firefox etc.

10. **Efficiency Requirements**

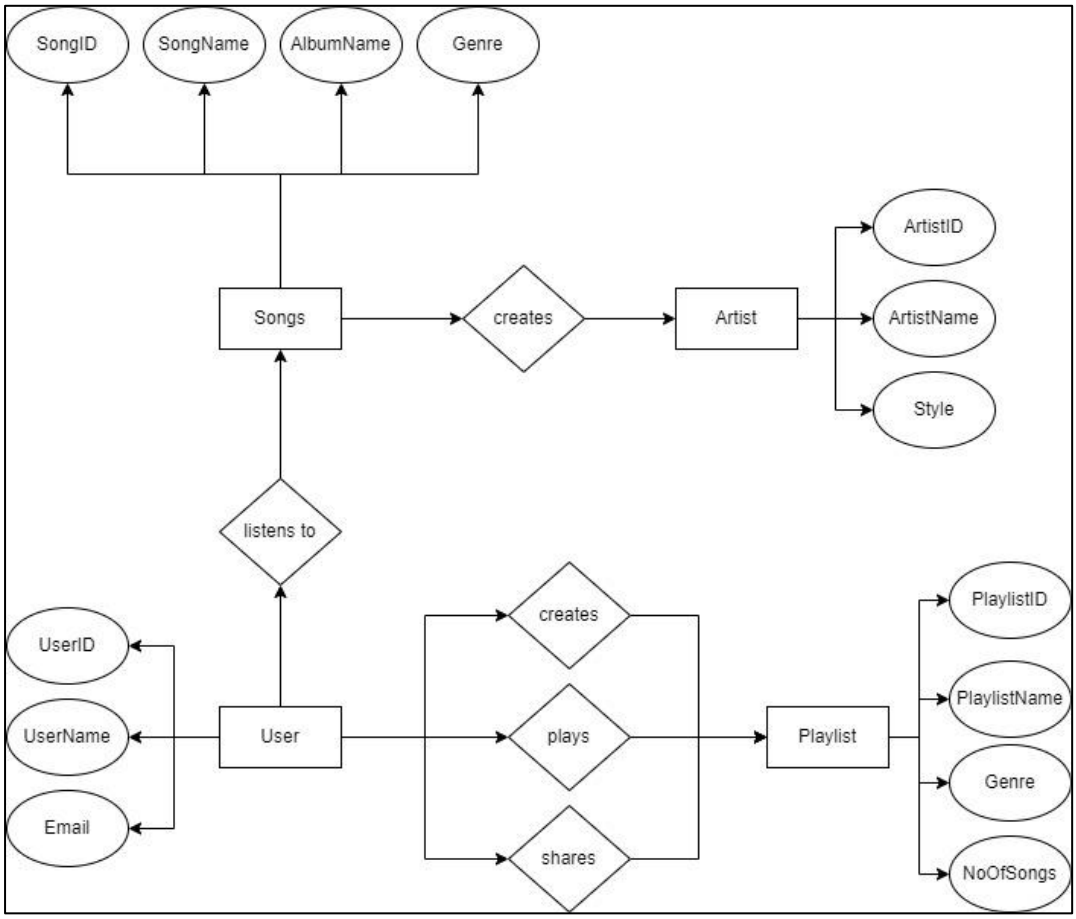
    It works with good efficiency.

11. **Interface Requirements**
    - Navigation: The website should have a clear and easy-to-use navigation system that allows users to access different functions and features.
    - Search: The website should have a search bar that allows users to search for songs, artists, albums, and playlists.

- Playlists: Users should be able to manage playlists easily.
- Playback controls: The website should provide clear and accessible playback controls, such as play, pause, skip, and volume control.
- User accounts: Users should be able to create and manage their accounts easily, with the ability to view their listening history, manage their preferences.

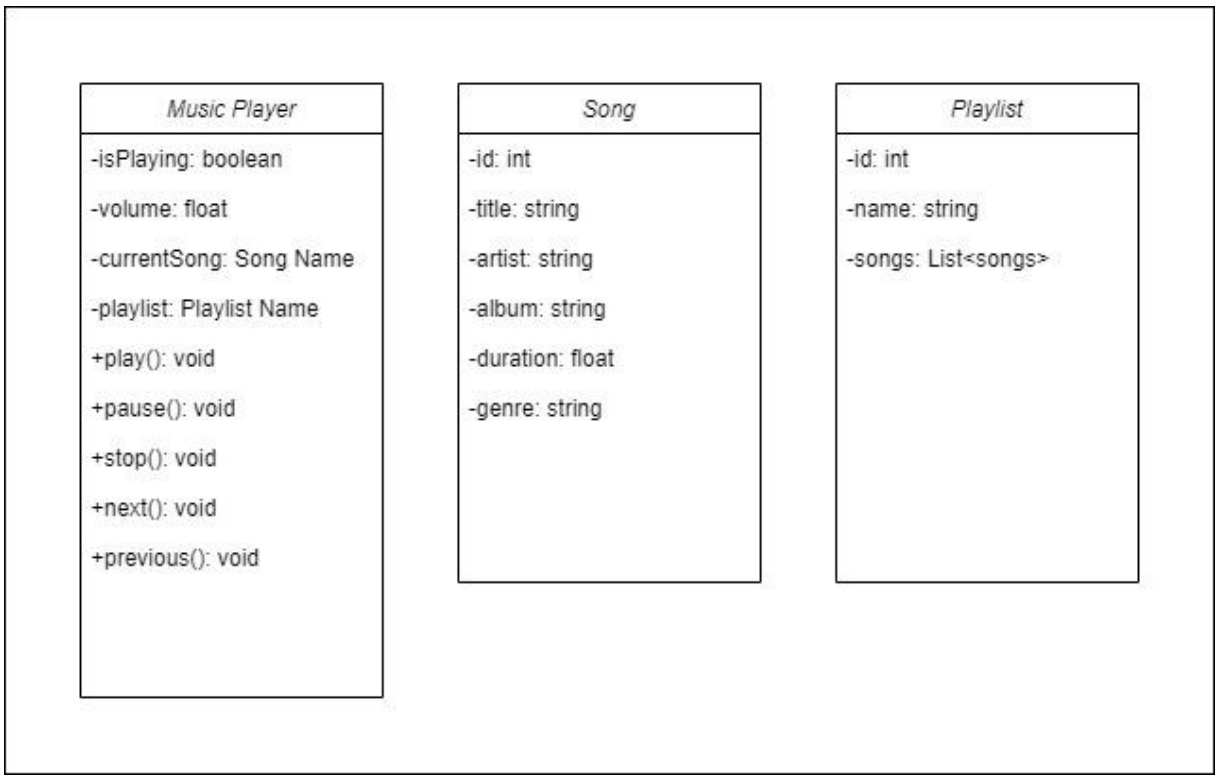
DIAGRAMS:

1. ER-Diagram:



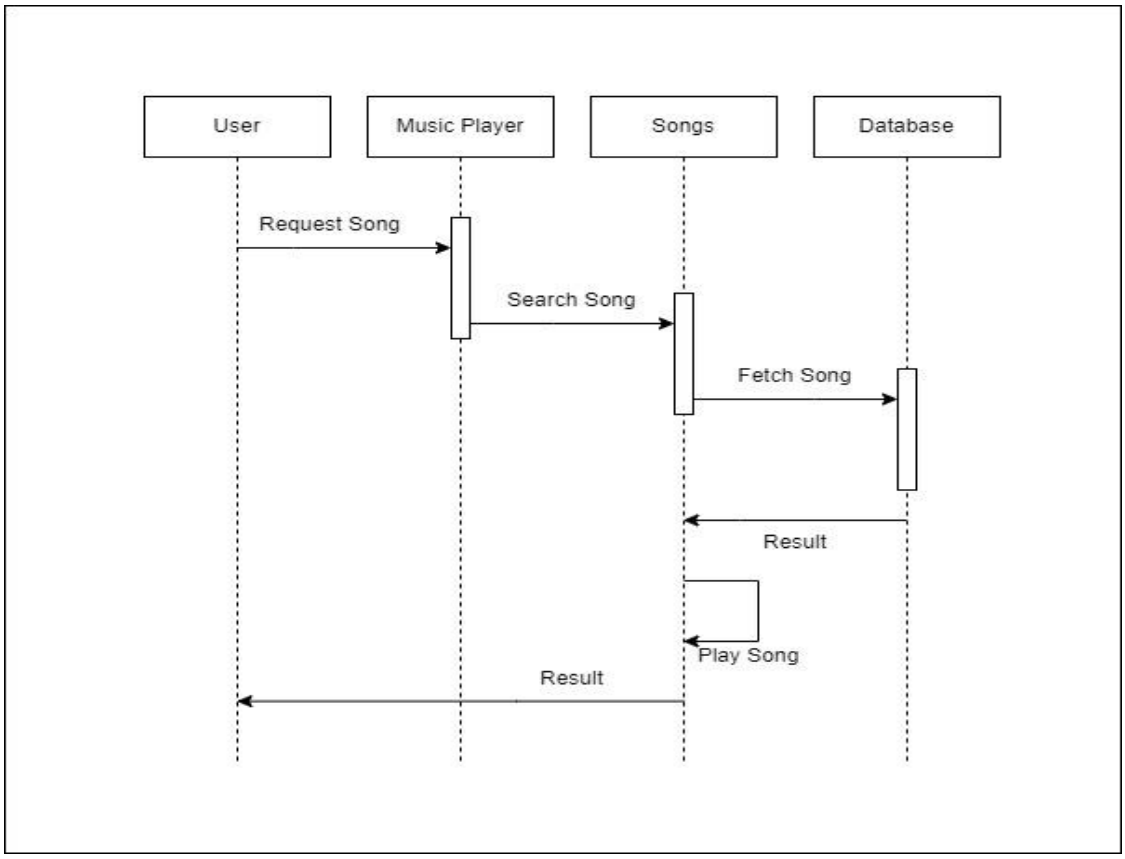
- An ER diagram of a music player would typically include entities such as Songs, Albums, Artists, and Playlists, each with their relevant attributes.
- These entities would be related to one another through relationships such as "belongs to," "performed by," "contains," and "has," with varying cardinalities depending on the specific requirements of the music player.
- Attributes associated with these entities could include unique IDs and user preferences.
- Additionally, other considerations such as users, subscription plans, and social features could also be included in the diagram to reflect the complexity of the music player.

2. Class Diagram:



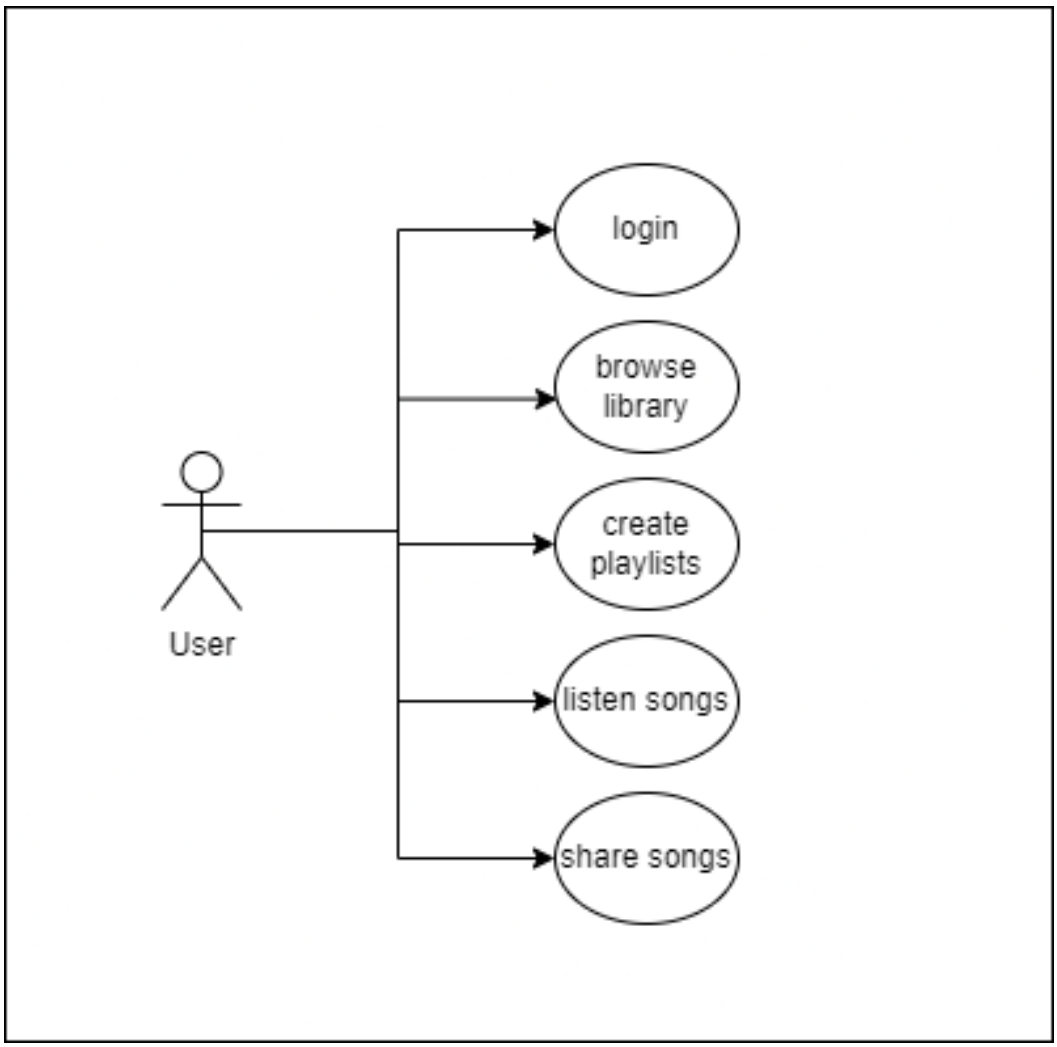
- A class diagram for a music player would typically include classes such as Music Player, Song, and Playlist, each with their relevant attributes and methods.
- Inheritance could be used to model the relationships between these classes, with a Playlist composed of multiple Song objects.
- Encapsulation could be used to restrict access to certain attributes and methods of each class.
- Other classes such as User, Subscription, and SocialFeatures could also be included in the diagram to reflect additional requirements.

3. Sequence Diagram



- A sequence diagram for a music player would typically depict actors such as the User, the Music Player application, Songs, and Database, with messages exchanged between these actors.
- The diagram would also show the timing and ordering of these messages, as well as any conditions or loops that could affect the sequence.
- Depending on the specific requirements of the music player, other actors and messages could be included in the diagram to reflect the complexity of the system.

4. Use Case Diagram:



- A use case diagram for a music player would typically depict actors such as the user, the music player application, and external systems or APIs, with use cases related to these actors through relationships such as "extends" and "includes".
- The use cases would include actions such as searching for songs, playing songs, creating and modifying playlists, and accessing external data sources.
- Depending on the specific requirements of the music player, other use cases and actors could be included in the diagram to reflect the complexity of the system.

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

This application will be a new hub for music lovers and a creative social media platform for music producers.

FUTURE SCOPE:

Integration of Social and Collaborative Features: Music is a highly social activity, and users often enjoy sharing their favourite songs and playlists with others. Therefore, future versions of the music player could integrate social and collaborative features such as the ability to create and share playlists with friends, follow other users and discover new music, or even collaborate with other users on a shared playlist.

Integration of Artificial Intelligence and Machine Learning: With the increasing sophistication of AI and machine learning technologies, music players could leverage these capabilities to provide more personalized and intelligent music recommendations, adapt to user preferences over time, and even generate new music based on user inputs.

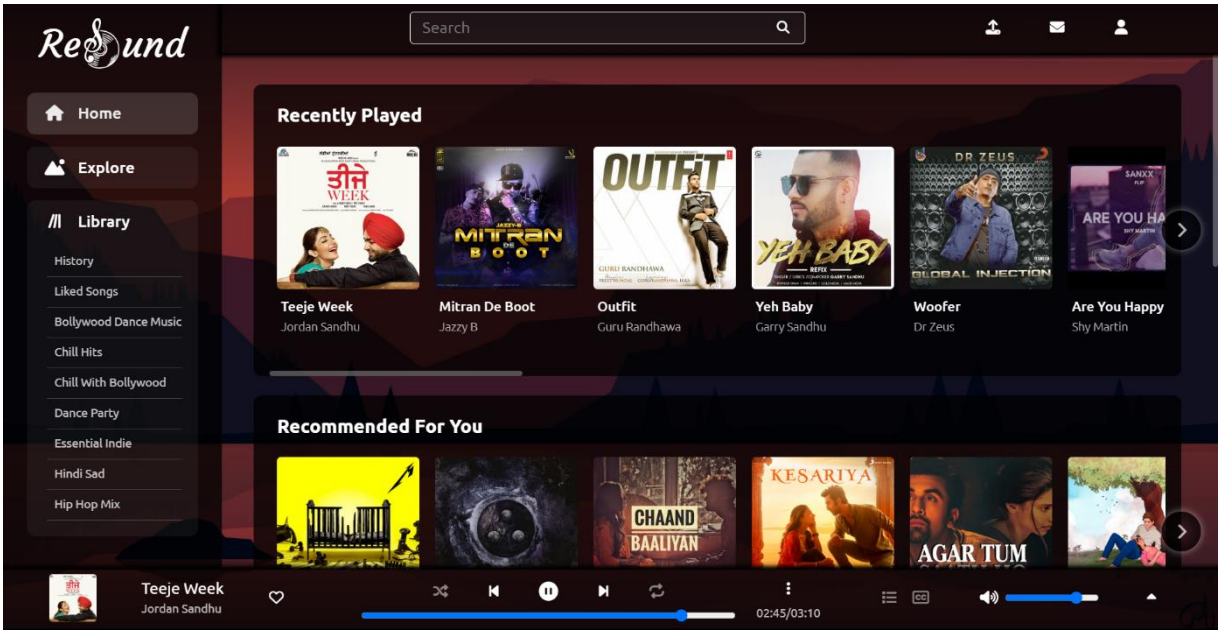
Augmented Reality and Virtual Reality Integration: Augmented reality and virtual reality technologies could be integrated into music players to create immersive and interactive music experiences. For example, users could experience concerts or live performances from the comfort of their homes, or even create their own virtual music environments

REFERENCES:

1. https://www.w3schools.com/
2. https://developer.mozilla.org/
3. https://www.geeksforgeeks.org/
4. https://stackoverflow.com/

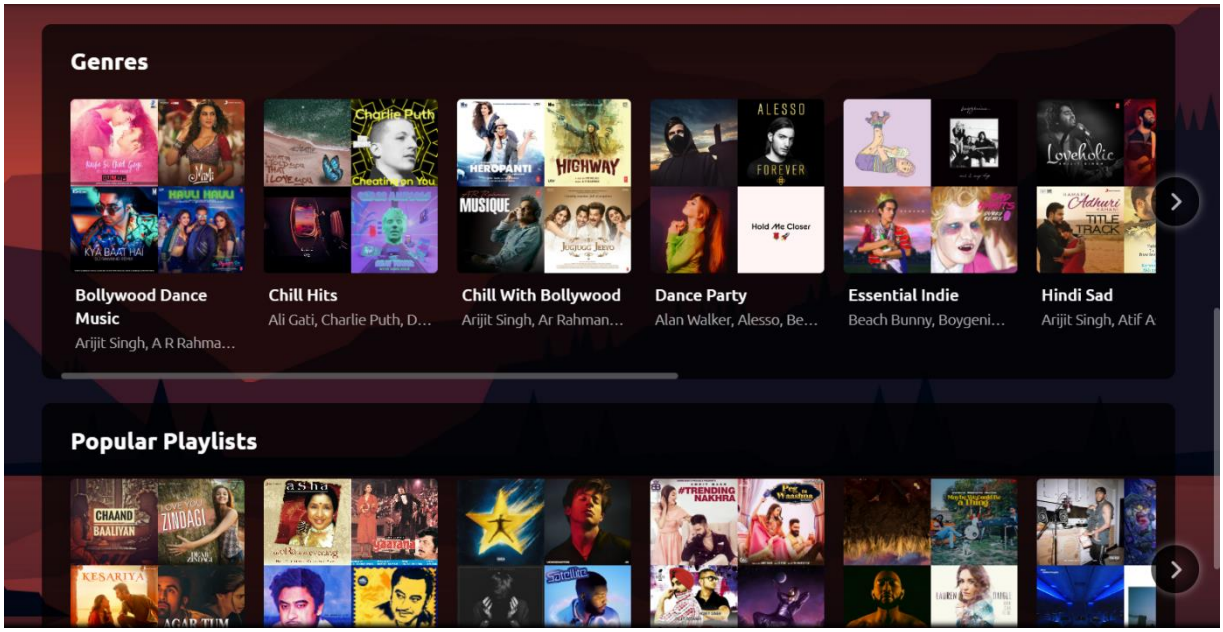
SNAPSHOTS

The Home Tab opens on website load and looks like this.



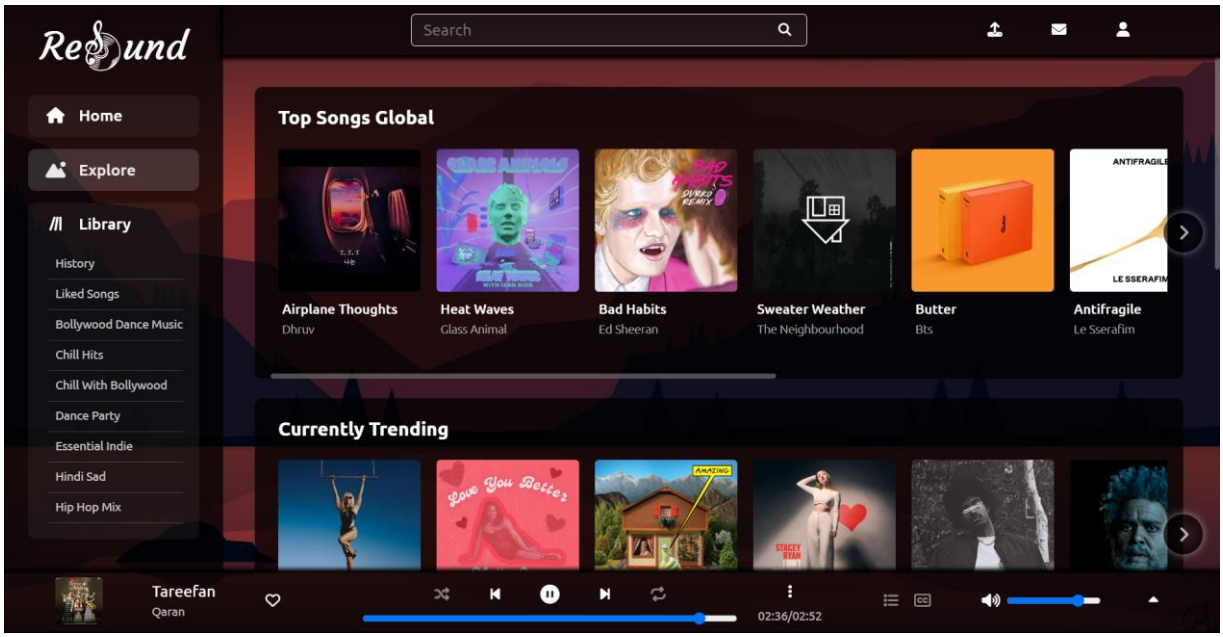
- The user's recently played songs and playlists, recommended songs, popular songs and genres are found in the Home Tab.
- There is a Side Navbar to switch between Home, Explore and Library Tab.
- The Header on top has the Search Bar for the user to search songs and artists on the platform, and it also has a Profile button on the right to Login/Sign Up or Logout.
- Clicking any song plays the song and adds the song to the recently played songs.
- Clicking any playlist opens the Playlist View and gives options to Play the playlist, Shuffle Play, Add to Library, and Add to Queue.

Genres and Popular Playlists are found in the Home Tab.



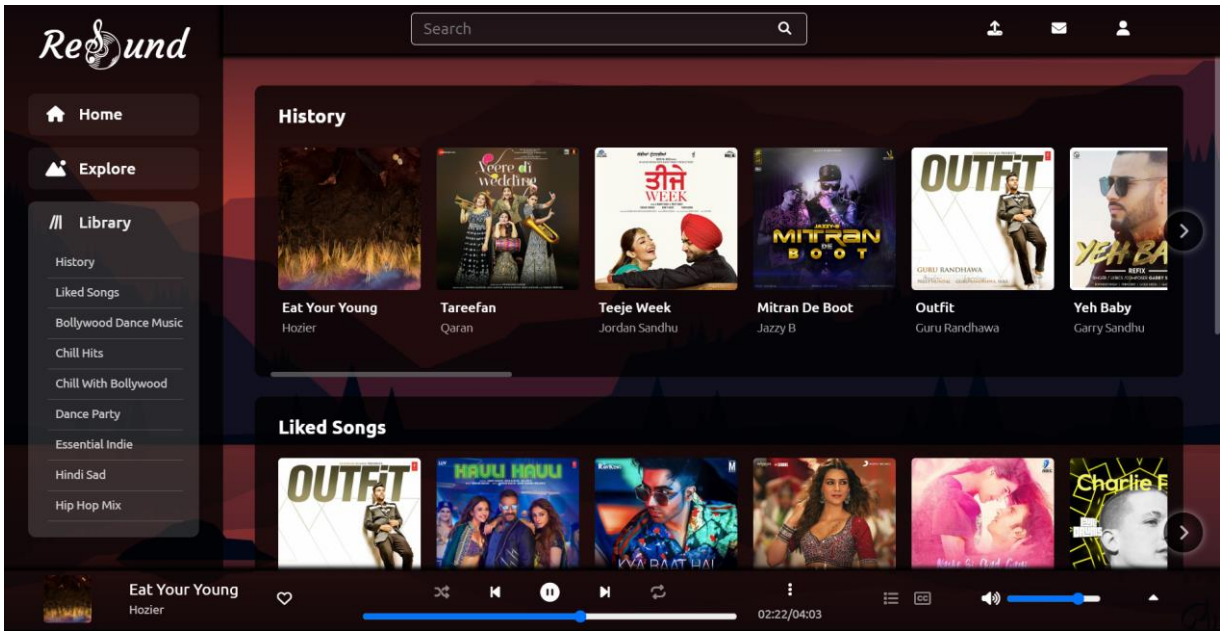
- You can also find multiple songs based on your preferred genre and various popular playlists in the Home Tab.
- You can click on the scroll buttons on the left and right of any category to scroll its content horizontally.
- Clicking the playlist cover opens the Playlist View, however, clicking the Play Icon Overlay on the cover directly plays the playlist from the Content Window.

The Explore Tab looks like this.



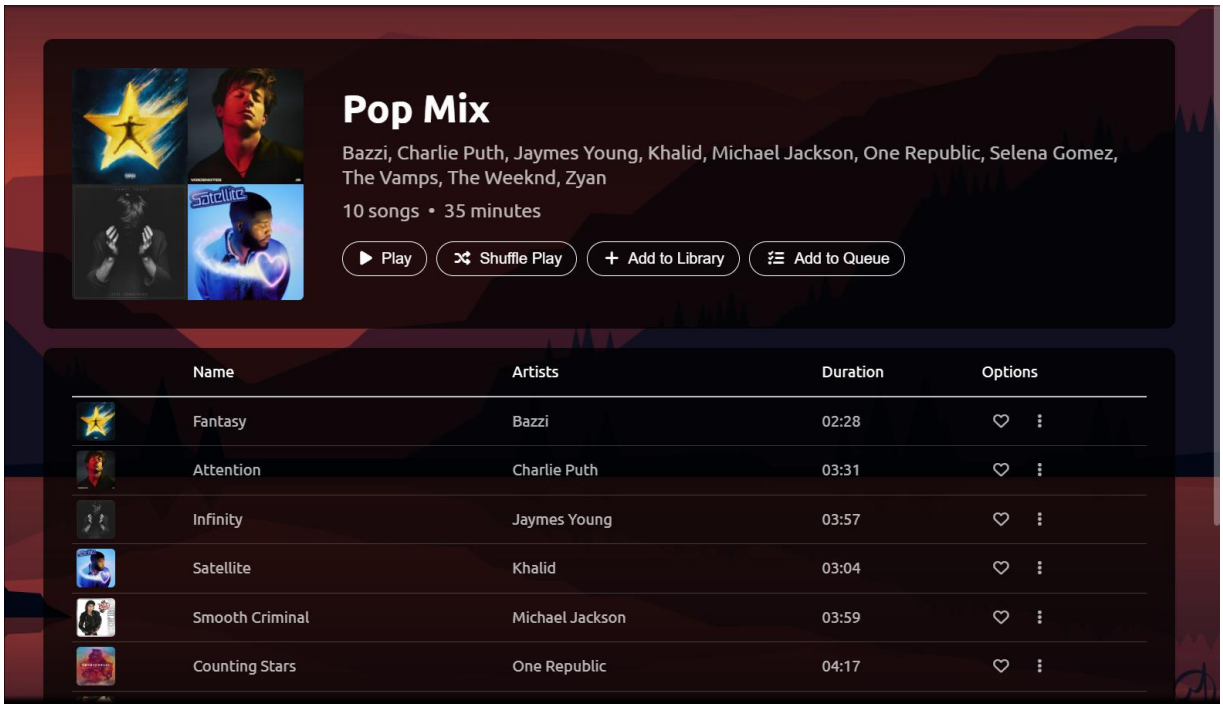
- Top songs on Global charts, Currently Trending Songs and Songs from Popular Artists are displayed in the Explore Tab.
- You can play songs, view playlists and scroll content in the same way as Home Tab on all tabs.
- The content on Explore Tab regular updates with time based on the user's preferences and their listening history.

The Library Tab looks like this.



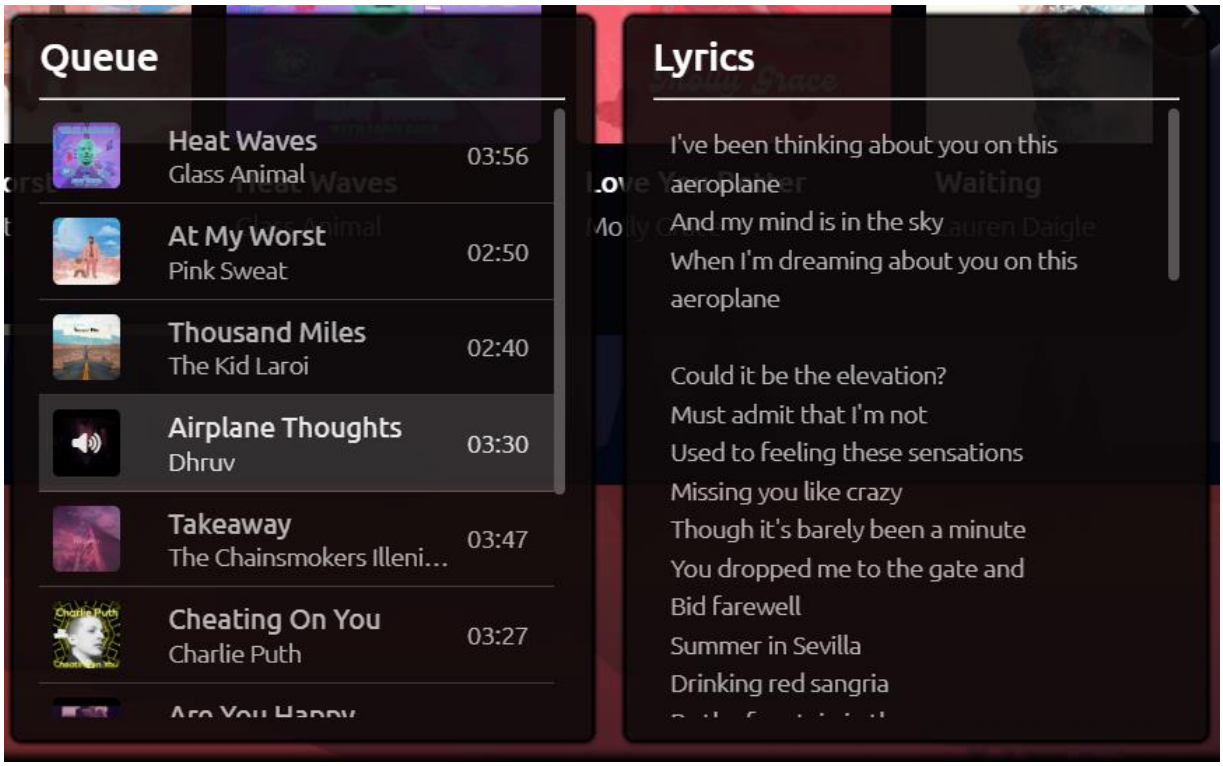
- The user's Listening History, Liked Songs and Saved Playlists are found in the Library Tab.
- As soon as the user plays a new song, the history gets updated and is displayed in the Library Tab.
- Same goes for Liked Songs i.e. as soon as the user likes or unlikes a song it gets added to the Liked Songs Playlist and can be viewed in the Library Tab.

Clicking any playlist opens the playlist view, which looks like this.



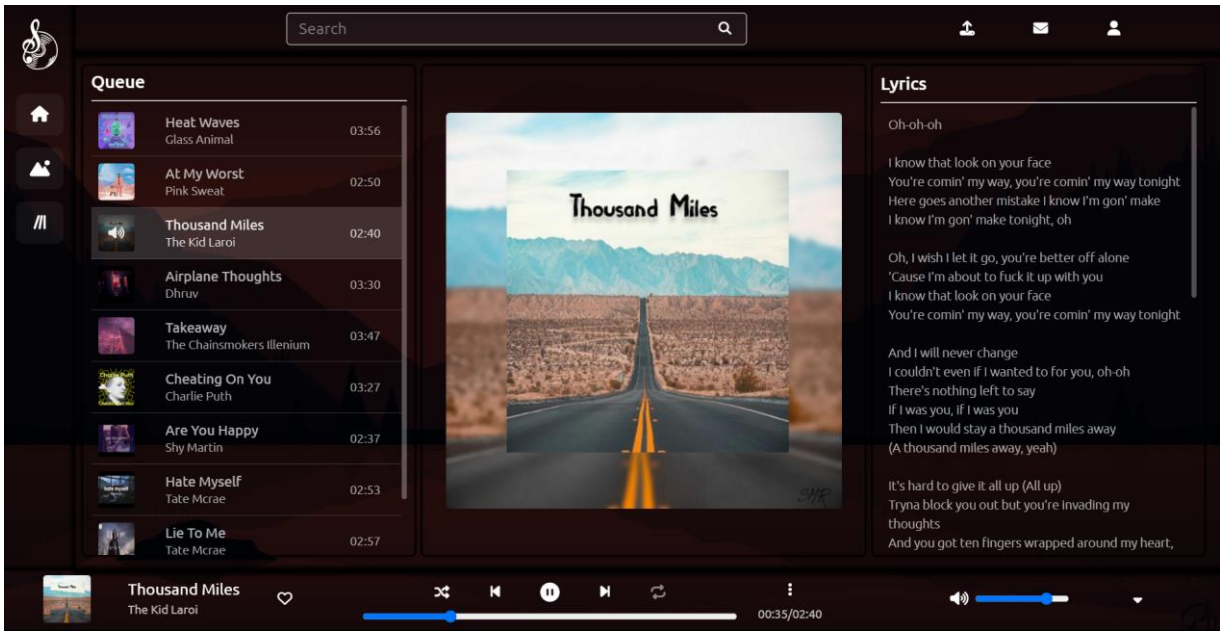
- This playlist view displays the Song Names along with their Covers, Artist Names, and Song Duration in the playlist contents.
- Options to Play the playlist, Shuffle Play, and Add to/Remove from Library are also available.
- The user can click on any song to play it, or like it to add it to their liked songs playlist or unlike to remove from the same.

Queue and Lyrics Floating Overlays show the queue and lyrics.



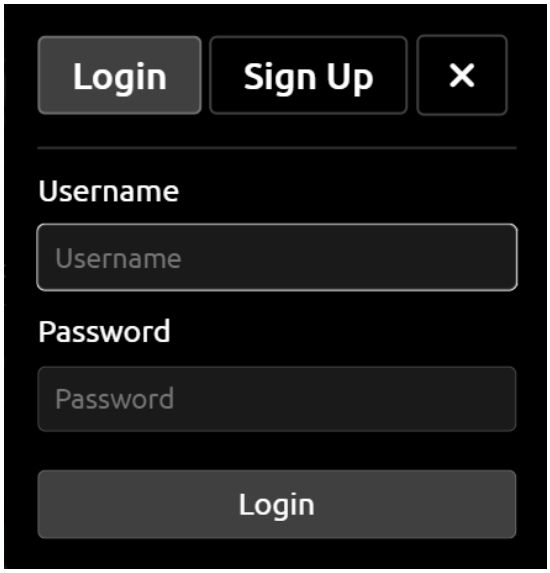
- These floating overlays can be toggled using the Queue and Lyrics buttons in the bottom player bar, or using the shortcut keys “Q” for Queue and “L” for Lyrics.
- The Queue Overlay shows the current playing queue songs with their covers, artists and duration.
- It also allows the user to play any other item in the queue by simply clicking it.
- The Lyrics Overlay simply displays the lyrics of the current playing song and gets updated as soon as a new song plays.

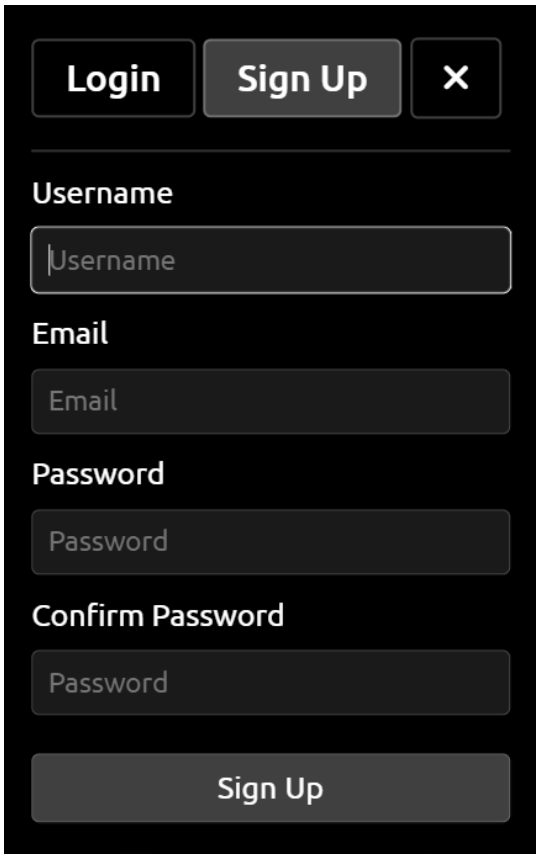
Extended Player for an immersive experience.



- The Extended Player can be toggled with the caret icon in bottom right and shows a larger view of the Queue, Album Cover and Lyrics of the playing song.
- The Queue Section in the Extended Player has the same functionality as the Floating Queue Overlay and allows the user to view the queue and play any other item in the queue.
- The Lyrics Section in the Extended Player shows a larger view of the lyrics to allow the user to easily read the lyrics while listening to their favourite songs.

Login/Sign Up Popups to Log In or Sign Up.





- The user can Log In to their account or Sign Up to a new account allowing them to Like songs, check their Listening History and Save Playlists to their Library.
- Signing Up requires the user to enter a new unique Username, a valid Email Address, Password and then Confirm Password, all of which have to be valid.
- Logging in simply requires the user to enter their Username and Password, and then click the Login Button.

PROJECT CODE:

HTML:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <link rel="stylesheet" type="text/css" href="style.css">
   <!-- FONT AWESOME CDN -->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">

   <!-- FAVICON LINKS -->
   <link rel="icon" type="image/x-icon" href="./media/favicon/white/favicon.ico">

   <link rel="apple-touch-icon" sizes="180x180" href="./media/favicon/white/apple-touch-icon.png">
   <link rel="icon" type="image/png" sizes="192x192" href="./media/favicon/white/android-chrome-
192x192.png">
   <link rel="icon" type="image/png" sizes="32x32" href="./media/favicon/white/favicon-32x32.png">
   <link rel="icon" type="image/png" sizes="16x16" href="./media/favicon/white/favicon-16x16.png">
   <link rel="manifest" href="./media/favicon/white/site.webmanifest">

   <!-- GOOGLE FONTS -->
   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
   <link

href="https://fonts.googleapis.com/css2?family=Courgette&family=Ubuntu:ital,wght@0,300;0,400;0,500;0,700;
1,300;1,400;1,500;1,700&display=swap"
      rel="stylesheet">

   <title>Re&ound | An Interactive Music Streaming Platform</title>
</head>

<body>

   <nav class="full-height">
```

```
<div id="nav-logo-container">
   <div id="logo-wrapper">

      <span class="nav-logo-text nav-shrinking-item">
         Re
      </span>
      <img id="nav-logo-image" src="./media/logos/logo_white.png" alt="logo">
      <span class="nav-logo-text nav-shrinking-item">
         und
      </span>

      <!-- Re□Ound -->
   </div>
</div>
<div id="nav-content">
   <div id="nav-home-button-container" class="nav-button-container">
      <div class="nav-button-icon-wrapper">
         <i class="fa-solid fa-house"></i>
      </div>
      <p class="nav-button-container-label nav-shrinking-item">
         Home
      </p>
   </div>
   <div id="nav-explore-button-container" class="nav-button-container">
      <div class="nav-button-icon-wrapper">
         <i class="fa-solid fa-mountain-sun"></i>
      </div>
      <p class="nav-button-container-label nav-shrinking-item">
         Explore
      </p>
   </div>
   <div id="nav-library-button-container" class="nav-button-container">
      <div class="nav-button-icon-wrapper">
         <i class="fa-solid fa-lines-leaning"></i>
      </div>
      <p class="nav-button-container-label nav-shrinking-item">
         Library
      </p>
      <div class="nav-library-content-items-wrapper nav-shrinking-item">
         <p class="nav-library-content-item">History</p>
         <p class="nav-library-content-item">Liked Songs</p>
         <p class="nav-library-content-item">2020 Favourites</p>
         <p class="nav-library-content-item">2021 Favourites</p>
         <p class="nav-library-content-item">Favourite Pop Music</p>
         <p class="nav-library-content-item">2022 Recap</p>
         <p class="nav-library-content-item">2021 Summer Recap</p>
         <p class="nav-library-content-item">My SuperMix 4</p>
```

```
            </div>
        </div>
    </div>
</nav>

<section class="right-window full-height">
    <header>
        <div id="header-left-empty-part"></div>
        <div id="header-middle-part">
            <div id="search-bar-wrapper">
                <input type="text" name="search" id="search-bar" placeholder="Search">
                <i class="fa-solid fa-magnifying-glass search-bar-icon"></i>
            </div>
            <div id="search-suggestions-wrapper" class="shrunk-element">
                <!-- <span class="search-suggestion">
                    <p class="search-suggestion-text">
                        Search History 1
                    </p>
                    <i class="fa-solid fa-magnifying-glass search-bar-icon"></i>
                </span>
                <span class="search-suggestion">
                    <p class="search-suggestion-text">
                        Search History 2
                    </p>
                    <i class="fa-solid fa-magnifying-glass search-bar-icon"></i>
                </span>
                <span class="search-suggestion">
                    <p class="search-suggestion-text">
                        Search History 3
                    </p>
                    <i class="fa-solid fa-magnifying-glass search-bar-icon"></i>
                </span>
                <span class="search-suggestion">
                    <p class="search-suggestion-text">
                        Search History 4
                    </p>
                    <i class="fa-solid fa-magnifying-glass search-bar-icon"></i>
                </span>
                <span class="search-suggestion">
                    <p class="search-suggestion-text">
                        Search History 5
                    </p>
                    <i class="fa-solid fa-magnifying-glass search-bar-icon"></i>
                </span> -->
            </div>
        </div>
        <div id="header-right-part">
```

```
            <!-- <button id="upload-button-container" class="header-right-part-button standard-button"
title="Upload">
                <i class="fa-solid fa-upload"></i>
            </button>
            <button id="messages-button-container" class="header-right-part-button standard-button"
title="Messages">
                <i class="fa-solid fa-envelope"></i>
            </button> -->
            <button id="profile-button-container" class="header-right-part-button standard-button" title="User
Profile">
                <i class="fa-solid fa-user"></i>
            </button>
            <div class="profile-dropdown-overlay shrunk-element invisible-element">
                <ul>
                    <li id="profile-dropdown-overlay-login-status">
                        Not logged in
                    </li>
                    <li id="profile-dropdown-overlay-login-logout-button">
                        Login / Sign Up
                    </li>
                </ul>
            </div>
        </div>
    </header>

    <section id="content-window">

        <!--
        <div id="recents-category" class="content-window-category">
            <button class="content-window-category-scroll-button content-window-category-scroll-left-button">
                <i class="fa-solid fa-chevron-left"></i>
            </button>
            <p class="content-window-category-label">Recently Played</p>
            <div class="content-window-category-items-wrapper">
                <div class="content-window-category-item">
                    <div class="content-window-category-item-cover-wrapper">
                        <img src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg"
alt="recents_item_cover" class="content-window-category-item-cover">
                        <div class="content-window-category-item-cover-overlay">
                            <div class="content-window-category-item-cover-overlay-icon-wrapper">
                                <i class="fa-solid fa-play"></i>
                            </div>
                        </div>
                    </div>
                    <div class="content-window-category-item-labels">
                        <p class="content-window-category-item-mainlabel">
                            Closer
```

```
                    </p>
                    <p class="content-window-category-item-sublabel">
                        The Chainsmokers
                    </p>
                </div>
            </div>
        </div>
        <button class="content-window-category-scroll-button content-window-category-scroll-right-button">
            <i class="fa-solid fa-chevron-right"></i>
        </button>
    </div>
    <div id="recommended-category" class="content-window-category">
        <button class="content-window-category-scroll-button content-window-category-scroll-left-button">
            <i class="fa-solid fa-chevron-left"></i>
        </button>
        <p class="content-window-category-label">Recommended For You</p>
        <div class="content-window-category-items-wrapper">
            <div class="content-window-category-item">
                <div class="content-window-category-item-cover-wrapper">
                    <img src="./media/sample_audio_and_covers/covers/imagine_dragons-birds-cover.jpeg"
alt="recommended_item_cover" class="content-window-category-item-cover">
                    <div class="content-window-category-item-cover-overlay">
                        <div class="content-window-category-item-cover-overlay-icon-wrapper">
                            <i class="fa-solid fa-play"></i>
                        </div>
                    </div>
                </div>
                <div class="content-window-category-item-labels">
                    <p class="content-window-category-item-mainlabel">
                        Birds
                    </p>
                    <p class="content-window-category-item-sublabel">
                        Imagine Dragons
                    </p>
                </div>
            </div>
        </div>
        <button class="content-window-category-scroll-button content-window-category-scroll-right-button">
            <i class="fa-solid fa-chevron-right"></i>
        </button>
    </div>
    <div id="genres-category" class="content-window-category">
        <button class="content-window-category-scroll-button content-window-category-scroll-left-button">
            <i class="fa-solid fa-chevron-left"></i>
        </button>
        <p class="content-window-category-label">Popular Genres</p>
        <div class="content-window-category-items-wrapper">
```

```
            <div class="content-window-category-item">
              <div class="content-window-category-item-cover-wrapper">

                <div class="content-window-category-item-cover">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                </div>

                <div class="content-window-category-item-cover-overlay">
                  <div class="content-window-category-item-cover-overlay-icon-wrapper">
                    <i class="fa-solid fa-play"></i>
                  </div>
                </div>
              </div>
              <div class="content-window-category-item-labels">
                <p class="content-window-category-item-mainlabel">
                  Pop
                </p>
              </div>
            </div>
          </div>
          <button class="content-window-category-scroll-button content-window-category-scroll-right-button">
            <i class="fa-solid fa-chevron-right"></i>
          </button>
        </div>
        <div id="moods-category" class="content-window-category">
          <button class="content-window-category-scroll-button content-window-category-scroll-left-button">
            <i class="fa-solid fa-chevron-left"></i>
          </button>
          <p class="content-window-category-label">Moods</p>
          <div class="content-window-category-items-wrapper">
            <div class="content-window-category-item">
              <div class="content-window-category-item-cover-wrapper">

                <div class="content-window-category-item-cover">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                  <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
```

```
                        <img src="./media/sample_audio_and_covers/covers/pop-cover.jpg"
alt="playlist_item_cover" class="content-window-category-item-cover-collage-image">
                    </div>

                    <div class="content-window-category-item-cover-overlay">
                        <div class="content-window-category-item-cover-overlay-icon-wrapper">
                            <i class="fa-solid fa-play"></i>
                        </div>
                    </div>
                </div>
                <div class="content-window-category-item-labels">
                    <p class="content-window-category-item-mainlabel">
                        Chill
                    </p>
                </div>
            </div>
        </div>
        <button class="content-window-category-scroll-button content-window-category-scroll-right-button">
            <i class="fa-solid fa-chevron-right"></i>
        </button>
    </div>
    -->

    <!--
    <div id="playlist-description">

        <div id="playlist-description-cover-wrapper">
            <div id="playlist-description-cover">
                <img src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg"
alt="playlist_cover" class="playlist-description-cover-collage-image">
                <img src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg"
alt="playlist_cover" class="playlist-description-cover-collage-image">
                <img src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg"
alt="playlist_cover" class="playlist-description-cover-collage-image">
                <img src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg"
alt="playlist_cover" class="playlist-description-cover-collage-image">
            </div>
        </div>

        <div id="playlist-description-info-and-buttons">
            <div id="playlist-description-info">
                <p id="playlist-info-title">Playlist Title</p>
                <p id="playlist-info-artists">Playlist Artists</p>
                <div id="playlist-info-count-and-duration">
                    <p id="playlist-info-song-count">
                        <span class="playlist-info-song-count-number">x</span>
                        <span class="playlist-info-song-count-label"> songs  •  </span>
```

```
            </p>
            <p id="playlist-info-duration">
              <span class="playlist-info-duration-number">y</span>
              <span class="playlist-info-duration-label"> minutes</span>
            </p>
          </div>
        </div>

        <div id="playlist-description-buttons">
          <button id="playlist-description-play-button" class="playlist-description-button">
            <i class="fa-solid fa-play playlist-description-button-icon"></i>
            <p class="playlist-description-button-label">Play</p>
          </button>
          <button id="playlist-description-shuffle-play-button" class="playlist-description-button">
            <i class="fa-solid fa-shuffle playlist-description-button-icon"></i>
            <p class="playlist-description-button-label">Shuffle Play</p>
          </button><br>
          <button id="playlist-description-add-to-library-button" class="playlist-description-button">
            <i class="fa-solid fa-plus playlist-description-button-icon"></i>
            <p class="playlist-description-button-label">Add to Library</p>
          </button>
          <button id="playlist-description-more-options-button" class="playlist-description-button">
            <i class="fa-solid fa-ellipsis-vertical playlist-description-button-icon"></i>
            <p class="playlist-description-button-label">More Options</p>
          </button>
        </div>
      </div>

    </div>

    <div id="playlist-content">

      <div id="playlist-content-header">
        <div class="playlist-content-cover-column"></div>
        <p class="playlist-content-name-column">Name</p>
        <p class="playlist-content-artists-column">Artists</p>
        <p class="playlist-content-duration-column">Duration</p>
        <p class="playlist-content-options-column">Options</p>
      </div>

      <hr>

      <div id="playlist-content-items">

        <div class="playlist-content-item">

          <div class="playlist-content-cover-column">
```

```
                    <img class="playlist-content-item-cover-image"
src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg" alt="playlist_item_cover">
                    <div class="content-item-cover-overlay-icon-wrapper">
                        <i class="fa-solid fa-play playlist-content-item-cover-image-overlay-icon"></i>
                    </div>
                </div>

                <p class="playlist-content-name-column">
                    Closer
                </p>
                <p class="playlist-content-artists-column">
                    The Chainsmokers
                </p>

                <p class="playlist-content-duration-column">
                    04:03
                </p>

                <div class="playlist-content-options-column">

                    <button class="playlist-content-item-like-button standard-button" title="Like">
                        <i class="fa-regular fa-heart"></i>
                        <i class="fa-solid fa-heart"></i>
                    </button>

                    <button class="playlist-content-item-options-button standard-button" title="Options">
                        <i class="fa-solid fa-ellipsis-vertical"></i>
                    </button>

                </div>

            </div>

            <div class="playlist-content-item">

                <div class="playlist-content-cover-column">
                    <img class="playlist-content-item-cover-image"
src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg" alt="playlist_item_cover">
                    <div class="content-item-cover-overlay-icon-wrapper">
                        <i class="fa-solid fa-play playlist-content-item-cover-image-overlay-icon"></i>
                    </div>
                </div>

                <p class="playlist-content-name-column">
                    Closer
                </p>
                <p class="playlist-content-artists-column">
```

```
                The Chainsmokers
              </p>

              <p class="playlist-content-duration-column">
                04:03
              </p>

              <div class="playlist-content-options-column">

                <button class="playlist-content-item-like-button standard-button" title="Like">
                  <i class="fa-regular fa-heart"></i>
                  <i class="fa-solid fa-heart"></i>
                </button>

                <button class="playlist-content-item-options-button standard-button" title="Options">
                  <i class="fa-solid fa-ellipsis-vertical"></i>
                </button>

              </div>

            </div>

          </div>
          -->

        </section>

        <div id="player-floating-part">

          <div id="floating-queue-overlay" class="player-floating-part-item-overlay shrunk-element">
            <p class="queue-lyrics-overlay-label">Queue</p>
            <hr>
            <div id="floating-queue-overlay-content-wrapper" class="queue-lyrics-content-wrapper">
              <!--
              <div class="queue-overlay-content-item">
                <div class="queue-content-item-cover-column">
                  <img class="queue-content-item-cover-image"
src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg" alt="queue_item_cover">
                  <div class="content-item-cover-overlay-icon-wrapper">
                    <i class="fa-solid fa-play queue-content-item-cover-overlay-icon"></i>
                    <i class="fa-solid fa-volume-high queue-content-item-cover-overlay-icon"></i>
                  </div>
                </div>
                <div class="queue-content-item-name-and-artists-column">
                  <p class="queue-content-item-name-row">
```

```
                        I Don't Wanna Live Forever (Fifty Shades Darker)
                    </p>
                    <p class="queue-content-item-artists-row">
                        ZAYN & Taylor Swift
                    </p>
                </div>
                <p class="queue-content-item-duration-column">
                    04:03
                </p>
            </div>
            -->
        </div>
    </div>

    <div id="floating-lyrics-overlay" class="player-floating-part-item-overlay shrunk-element">
        <p class="queue-lyrics-overlay-label">Lyrics</p>
        <hr>
        <p id="floating-lyrics-overlay-content-wrapper" class="queue-lyrics-content-wrapper lyrics-content-
wrapper"></p>
    </div>

</div>

<div class="player-extended-overlay shrunk-element">
    <div id="player-extended-queue" class="player-extended-overlay-part">
        <p class="queue-lyrics-overlay-label">Queue</p>
        <hr>
        <div id="player-extended-queue-content-wrapper" class="queue-lyrics-content-wrapper">
            <!--
            <div class="queue-overlay-content-item">
                <div class="queue-content-item-cover-column">
                    <img class="queue-content-item-cover-image"
src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg" alt="queue_item_cover">
                    <div class="content-item-cover-overlay-icon-wrapper">
                        <i class="fa-solid fa-play queue-content-item-cover-overlay-icon"></i>
                    </div>
                </div>
                <div class="queue-content-item-name-and-artists-column">
                    <p class="queue-content-item-name-row">
                        I Don't Wanna Live Forever (Fifty Shades Darker)
                    </p>
                    <p class="queue-content-item-artists-row">
                        ZAYN & Taylor Swift
                    </p>
                </div>
                <p class="queue-content-item-duration-column">
                    04:03
```

```
                </p>
              </div>
               -->
            </div>
          </div>
          <div id="player-extended-cover" class="player-extended-overlay-part">
            <!-- <div id="player-extended-cover-image-wrapper"> -->
              <!-- <img id="player-extended-cover-image"
src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg" alt="song_cover_image"> --
>
              <!-- <img id="player-extended-cover-image" src="" alt="song_cover_image"> -->
            <!-- </div> -->
          </div>
          <div id="player-extended-lyrics" class="player-extended-overlay-part">

            <p class="queue-lyrics-overlay-label">Lyrics</p>
            <hr>
            <p id="player-extended-lyrics-content-wrapper" class="queue-lyrics-content-wrapper lyrics-content-
wrapper"></p>
          </div>
        </div>

    </section>

    <div class="player inactive-player">

        <div id="player-left-part">
          <div id="player-song-cover">
            <!-- <img id="player-song-cover-image"
src="./media/sample_audio_and_covers/covers/the_chainsmokers-closer-cover.jpg" alt="song_cover_image"> --
>
            <img id="player-song-cover-image" src="" alt="song_cover_image">
          </div>

          <div id="player-song-info">
            <!-- <p id="player-song-name">Closer</p><br>
            <p id="player-song-artist">The Chainsmokers</p> -->
            <!-- <p id="player-song-name">I Don't Wanna Live Forever</p><br> -->
            <!-- <p id="player-song-artist">Taylor Swift and Zayn Malik</p> -->
            <p id="player-song-name"></p><br>
            <p id="player-song-artist"></p>
          </div>

          <button id="player-left-part-like-button-container" class="standard-button" title="Like">
            <i class="fa-regular fa-heart"></i>
            <!-- <i class="fa-regular fa-heart"></i> -->
          </button>
```

```
</div>

<div id="player-middle-part">
   <div id="player-middle-part-main">
      <div id="player-middle-part-controls">
         <button id="shuffle-button-container"
            class="standard-button" title="Shuffle">
            <i class="fa-solid fa-shuffle inactive-button"></i>
         </button>

         <button id="previous-song-button-container"
            class="standard-button" title="Play Previous Song">
            <i class="fa-solid fa-backward-step"></i>
         </button>
         <button id="play-pause-button-container"
            class="standard-button" title="Play">
            <i class="fa-solid fa-circle-play"></i>
            <!-- <i class="fa-solid fa-circle-pause"></i> -->
         </button>
         <button id="next-song-button-container"
            class="standard-button" title="Play Next Song">
            <i class="fa-solid fa-forward-step"></i>
         </button>

         <button id="repeat-button-container"
            class="standard-button" title="Repeat">
            <i class="fa-solid fa-1 shrunk-element"></i>
            <i class="fa-solid fa-repeat inactive-button"></i>
         </button>
      </div>

      <input type="range" name="player-seek-bar" id="player-seek-bar" value="0">
   </div>

   <div id="player-middle-part-right">
      <button id="player-options-button-container" class="standard-button" title="Options">
         <i class="fa-solid fa-ellipsis-vertical"></i>
      </button>
      <div id="player-middle-part-timestamp">
         <span id="audio-current-duration"></span>/<span id="audio-total-duration">04:03</span>
      </div>
   </div>

</div>

<div id="player-right-part">
   <div id="player-right-part-main-controls">
```

```
            <button id="queue-button-container" class="standard-button" title="Queue">
                <i class="fa-solid fa-list inactive-button"></i>
            </button>
            <button id="lyrics-button-container" class="standard-button" title="Lyrics">
                <!-- <i class="fa-solid fa-closed-captioning"></i> -->
                <i class="fa-regular fa-closed-captioning inactive-button"></i>
            </button>
        </div>

        <div id="player-right-part-volume-controls">
            <button id="volume-button-container" class="standard-button" title="Mute">
                <i id="player-right-part-volume-button" class="fa-solid fa-volume-high"></i>
            </button>

            <input type="range" name="volume-seek-bar" id="volume-seek-bar" value="80" title="Volume">
        </div>

        <button id="player-extend-button-container" class="standard-button" title="Extend Player">
            <i class="fa-solid fa-caret-up"></i>
            <!-- <i class="fa-solid fa-caret-down"></i> -->
        </button>
    </div>

    <audio id="player-audio-controls">
        <!-- <source
src="./media/sample_audio_and_covers/audio/renamed/Closer___The_Chainsmokers_ft__Halsey_320_PagalWo
rld_.mp3" type="audio/mpeg"> -->
        <source src="" type="audio/mpeg">
    </audio>

</div>

<div id="body-wallpaper">
    <img class="body-wallpaper-image" src="./media/wallpapers/3046432.jpg" alt="background_wallpaper">
    <div class="body-wallpaper-overlay"></div>
</div>

<div class="login-popup invisible-element">
    <div class="login-popup-fade-bg"></div>
    <div class="login-popup-box">
        <div class="login-popup-tab-buttons">
            <div id="login-popup-login-tab-button" class="login-popup-tab-button login-popup-tab-active-
button">
                Login
            </div>
            <div id="login-popup-signup-tab-button" class="login-popup-tab-button">
                Sign Up
```

```
            </div>
            <i id="login-popup-close-button" class="fa-solid fa-times login-popup-tab-button"></i>
        </div>
        <hr>
        <div class="login-popup-login-tab invisible-element shrunk-element">
            <form class="login-popup-tab-form">

                <p class="login-popup-input-label">Username</p>
                <input type="text" name="username" class="login-popup-username-input"
placeholder="Username">
                <p class="login-popup-username-caution-text login-popup-caution-text display-none"></p>

                <p class="login-popup-input-label">Password</p>
                <input type="password" name="password" class="login-popup-password-input"
placeholder="Password">
                <p class="login-popup-password-caution-text login-popup-caution-text display-none"></p>

            </form>
            <button id="login-popup-submit-login-button" class="login-popup-submit-button">
                Login
            </button>
        </div>
        <div class="login-popup-signup-tab invisible-element shrunk-element">
            <form class="login-popup-tab-form">

                <p class="login-popup-input-label">Username</p>
                <input type="text" name="username" class="login-popup-username-input"
placeholder="Username">
                <p class="login-popup-username-caution-text login-popup-caution-text display-none"></p>

                <p class="login-popup-input-label">Email</p>
                <input type="email" name="email" class="login-popup-email-input" placeholder="Email">
                <p class="login-popup-email-caution-text login-popup-caution-text display-none"></p>

                <p class="login-popup-input-label">Password</p>
                <input type="password" name="password" class="login-popup-password-input"
placeholder="Password">
                <p class="login-popup-password-caution-text login-popup-caution-text display-none"></p>

                <p class="login-popup-input-label">Confirm Password</p>
                <input type="password" name="confirm-password" class="login-popup-confirm-password-input"
placeholder="Password">
                <p class="login-popup-confirm-password-caution-text login-popup-caution-text display-none"></p>

            </form>
            <button id="login-popup-submit-signup-button" class="login-popup-submit-button">
                Sign Up
```

```
            </button>
          </div>
        </div>
    </div>

    <!-- <div class="floating-notification">
        <i class="fa-solid fa-square-check floating-notification-icon"></i>
        <p class="floating-notification-text">
            Added playlist to library
        </p>
    </div> -->

    <!-- IMPORTING SCRIPTS -->
    <script type="module" src="./script.js"></script>
</body>

</html>
```

CSS:

```
:root{
    --header-height: 64px;
    --player-height: 75px;
    --nav-width: 18%;
    --nav-min-width: 250px;
    --nav-max-width: 350px;
    --content-window-item-background-color: rgba(0, 0, 0, 0.5);
    --content-window-category-item-cover-wrapper-side: 180px;
    --playlist-description-cover-side: 240px;
    --sub-label-color: rgba(255, 255, 255, 0.7);
    --hover-bg-color-light: rgba(255, 255, 255, 0.25);
    --item-border-bottom-color: rgba(255, 255, 255, 0.2);
    --panels-bg-color: rgba(0, 0, 0, 0.75);
    --panels-box-shadow-color: black;
    --panels-box-shadow: 0 0 5px 5px var(--panels-box-shadow-color);
    --overlay-box-shadow: 0 0 2px 2px var(--panels-box-shadow-color);
    --buttons-box-shadow: 0 0 10px 7px var(--hover-bg-color-light) inset;
    --hover-box-shadow-color: rgba(0, 0, 0, 0.5);
    /* --floating-panels-bg-color: rgba(0, 0, 0, 0.90); */
    --floating-panels-bg-color: rgb(15, 10, 10, 0.9);
    --playlist-page-bg-color: var(--panels-bg-color);
    --playlist-content-item-cover-side: 2rem;
    --floating-queue-content-item-cover-side: 1.9rem;
```

```
    --player-extended-queue-content-item-cover-side: 2.3rem;
    --transition-properties: all 0.2s;
    /* --transition-properties: none; */
}

/* SCROLLBAR CUSTOMIZATION */
::-webkit-scrollbar {
        width: 7px;
    height: 7px;
    border-radius: 4px;

}
::-webkit-scrollbar-thumb {
        width: 7px;
    height: 7px;
        background: rgba(255, 255, 255, 0.3);
        border-radius: 4px;
}
::-webkit-scrollbar-thumb:hover {
        background: rgba(255, 255, 255, 0.5);
}

html, body{
    height: 100%;
    width: 100%;
    /* min-width: 900px; */
    font-size: 20px;
    font-family: 'Ubuntu', sans-serif;
    box-sizing: border-box;
    position: relative;
}
*, *:before, *:after{
    margin: 0;
    padding: 0;
    text-decoration: none;
    box-sizing: inherit;
        transition: var(--transition-properties);
    scroll-behavior: smooth;
}

/* TAGS */
p{
    display: inline-block;
    margin: 0;
    padding: 0;
}
a{
```

```
    color: white;
}
input{
    font-family: inherit;
}
input[type='range']{
    cursor: pointer;
    border-radius: 0.5rem;
}
input:focus-visible{
    outline: 1px solid white;
}
input[type='range']:focus-visible{
    outline-offset: 0.25rem;
}
button{
    background-color: transparent;
    border: none;
    outline: none;
    color: inherit;
    cursor: pointer;
    font-size: inherit;
}
button:focus-visible{
    /* outline: 1px solid white; */
    box-shadow: var(--buttons-box-shadow);
}
hr{
    width: 100%;
}
.standard-button{
    position: relative;
    height: 2rem;
    width: 2rem;
    border-radius: 1rem;
    display: inline-flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    font-size: 0.9rem;
}
.standard-button:hover{
    box-shadow: var(--buttons-box-shadow);
}


/* WALLPAPER */
```

```
.body-wallpaper-image{
    height: 100%;
    width: 100%;
    object-fit: cover;
    position: fixed;
    top: 0;
    left: 0;
    z-index: -100;
}
.body-wallpaper-overlay{
    height: 100%;
            width: 100%;
            background-color: rgb(0, 0, 0, 0.25);
            position: fixed;
            top: 0;
    left: 0;
            z-index: -50;
            /* overflow-y: auto; */
}



/* INSIDE BODY */
nav{
    height: calc(100% - var(--player-height));
    width: var(--nav-width);
    min-width: var(--nav-min-width);
    max-width: var(--nav-max-width);
    background-color: var(--panels-bg-color);
    color: white;
    position: fixed;
    top: 0;
    left: 0;
    z-index: 10;
}
#nav-logo-container{
    height: calc(1rem + var(--header-height));
    display: flex;
    align-items: center;
    justify-content: space-around;
    font-size: 2rem;
    padding-top: 1rem;
}
#logo-wrapper{
    height: 100%;
    display: flex;
    align-items: center;
```

```
    justify-content: center;
    font-family: 'Courgette', cursive;
    font-size: 2.4rem;
    cursor: pointer;
}
#nav-logo-image{
    height: 64px;
    padding: 0 3px 0 4px;
}
#nav-content{
    height: calc(100% - 1rem - var(--header-height));
    overflow-y: hidden;
    padding: 0.75rem;
}
#nav-content:hover{
    overflow-y: auto;
}
.nav-button-container{
    background-color: rgba(255, 255, 255, 0.05);
    border-radius: 0.5rem;
    text-align: left;
    cursor: pointer;
    margin: 0.75rem 0.75rem 0;
    padding: 0.75rem 1rem;
}
.nav-button-container:hover{
    background-color: var(--hover-bg-color-light);
}
.nav-button-container-shrunk{
    margin: 0.75rem 0 0;
    padding: 0.75rem 0.5rem;
    display: flex;
    align-items: center;
    justify-content: center;
}
.nav-button-icon-wrapper{
    display: inline-block;
    width: 1.5rem;
    text-align: center;
}
.nav-button-container-label{
    font-weight: 500;
    margin-left: 0.5rem;
}
.nav-library-content-items-wrapper{
    margin: 1rem 0 0.25rem 0.25rem;
    max-height: 17.3rem;
```

```
    overflow-y: hidden;
}
#nav-library-button-container:hover{
    padding-right: 0.75rem;
    transition: none 0s;
}
#nav-library-button-container:hover .nav-library-content-items-wrapper{
    overflow-y: auto;
}
.nav-library-content-item{
    width: 100%;
    color: rgba(255, 255, 255, 0.75);
    font-size: 0.75rem;
    text-align: left;
    display: block;
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
    border-bottom: 1px solid var(--item-border-bottom-color);
    padding: 0.5rem 0 0.5rem 0.5rem;
}
.nav-library-content-item:hover{
    color: white;
    background-color: rgba(0, 0, 0, 0.35);
}

.right-window{
    height: calc(100% - var(--player-height));
    width: calc(100% - var(--nav-width));
    min-width: calc(100% - var(--nav-max-width));
    max-width: calc(100% - var(--nav-min-width));
    position: fixed;
    top: 0;
    right: 0;
    bottom: var(--player-height);
}


/* INSIDE RIGHT WINDOW */
header{
    height: var(--header-height);
    width: inherit;
    min-width: inherit;
    max-width: inherit;
    position: fixed;
    top: 0;
    right: 0;
```

```
    z-index: 20;
    background-color: var(--panels-bg-color);
    box-shadow: var(--panels-box-shadow);
    display: flex;
    align-items: center;
    justify-content: space-between;
    color: white;
    padding: 0 4%;
    transition: none;
}
#header-left-empty-part{
    height: 100%;
    width: 10%;
}
#header-middle-part{
    width: 55%;
    min-width: 300px;
    display: flex;
    align-items: center;
    justify-content: center;
    position: relative;
    font-weight: 300;
}
#search-bar-wrapper{
    width: 80%;
    position: relative;
    display: flex;
    align-items: center;
}
#search-bar{
    width: 100%;
    color: white;
    font-size: 1rem;
    padding: 0.4rem 2.5rem 0.4rem 0.75rem;
    border-radius: 0.25rem;
    background-color: rgba(255, 255, 255, 0.05);
    border: 1px solid white;
    position: relative;
}
#search-suggestions-wrapper{
    width: calc(80% - 1.5rem);
    font-size: 0.9rem;
    position: absolute;
    top: 2.1rem;
    background-color: var(--floating-panels-bg-color);
    border-radius: 0 0 0.25rem 0.25rem;
}
```

```
.search-suggestion{
    display: flex;
    align-items: center;
    width: 100%;
    padding: 0.75rem 2.5rem 0.75rem 1rem;
    cursor: pointer;
    position: relative;
}
.search-suggestion-text{
    width: 100%;
}
.search-suggestion:hover{
    background-color: var(--hover-bg-color-light);
}
.search-bar-icon{
    position: absolute;
    font-size: 0.8em;
    right: 1rem;
    color: white;
}
#header-right-part{
    width: 25%;
    margin: 0 0 0 10%;
    display: flex;
    align-items: center;
    /* justify-content: space-evenly; */

    /* temporarily disabling upload and messaging buttons and changing justify-content to right */
    justify-content: right;
    padding-right: 1rem;
}
.profile-dropdown-overlay{
    position: absolute;
    top: calc(var(--header-height) + 0.5rem);
    right: 7.5%;
    width: 10rem;
    background-color: var(--floating-panels-bg-color);
    box-shadow: var(--overlay-box-shadow);
    padding: 0.35rem;
    border-radius: 0.25rem 0 0.25rem 0.25rem;
    font-weight: 400;
    font-size: 0.75rem;
}
.profile-dropdown-overlay ul li{
    width: 100%;
    padding: 0.5rem;
    /* border-top: 1px solid var(--item-border-bottom-color); */
```

```
    border-bottom: 1px solid var(--item-border-bottom-color);
    list-style: none;
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
}
#profile-dropdown-overlay-login-status{
    cursor: default
}
.profile-dropdown-overlay ul li:not(#profile-dropdown-overlay-login-status):hover{
    background-color: var(--hover-bg-color-light);
}
.login-popup-fade-bg{
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
    z-index: 20;
}
.login-popup-box{
    width: 350px;
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background-color: var(--floating-panels-bg-color);
    background-color: black;
    box-shadow: var(--overlay-box-shadow);
    padding: 1rem;
    border-radius: 0.25rem;
    z-index: 21;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-evenly;
    font-size: 1.2rem;
    color: white;
    overflow-y: auto;
    max-height: calc(100% - var(--header-height) - var(--player-height) - 2rem);
}
.login-popup-tab-buttons{
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: space-between;
```

```
    font-size: 1.2rem;
    font-weight: 500;
}
.login-popup-tab-button{
    flex-grow: 1;
    cursor: pointer;
    padding: 0.5rem;
    border-radius: 0.25rem;
    text-align: center;
    margin: 0 0.25rem 0 0;
    border: 2px solid var(--hover-bg-color-light);
}
.login-popup-tab-active-button{
    background-color: var(--hover-bg-color-light);
}
:not(.login-popup-tab-active-button).login-popup-tab-button:hover{
    background-color: var(--hover-bg-color-light);
}
#login-popup-close-button{
    height: 2.64rem;
    display: flex;
    align-items: center;
    justify-content: center;
}
.login-popup-box hr{
    width: 100%;
    border: 1px solid var(--item-border-bottom-color);
    margin: 1rem 0 0;
}
.login-popup-input-label{
    font-weight: 400;
    margin-top: 0.75rem;
    font-size: 1rem;
}
.login-popup-box input{
    width: 100%;
    padding: 0.5rem;
    border-radius: 0.25rem;
    border: 1px solid var(--item-border-bottom-color);
    background-color: rgba(255, 255, 255, 0.1);
    color: white;
    font-size: 0.9rem;
    font-weight: 400;
    margin: 0.5rem 0 0;
}
.login-popup-caution-text{
    font-size: 0.8rem;
```

```
    padding-top: 0.25rem;
    color: rgba(255, 32, 32, 0.75);
    width: 100%;
}
.login-popup-submit-button{
    width: 100%;
    padding: 0.5rem;
    border-radius: 0.25rem;
    border: 1px solid var(--item-border-bottom-color);
    background-color: rgba(255, 255, 255, 0.25);
    color: white;
    font-family: inherit;
    font-size: 1rem;
    font-weight: 400;
    margin: 1.25rem 0 0;
}
.login-popup-submit-button:hover{
    background-color: rgba(255, 255, 255, 0.35);
}

/* RIGHT CONTENT WINDOW */
#content-window{
    position: absolute;
    top: var(--header-height);
    right: 0;
    bottom: var(--player-height);
    height: calc(100% - var(--header-height));
    width: 100%;
    min-width: 100%;
    max-width: 100%;
    overflow-y: auto;
    display: flex;
    align-items: center;
    flex-direction: column;
    color: white;
}

/* GENERAL CONTENT WINDOW STYLES */
.content-window-category{
    width: calc(100% - 4rem);
    flex-shrink: 0;
    background-color: var(--panels-bg-color);
    margin-top: 1.25rem;
    border-radius: 0.5rem;
    padding: 1rem 1rem 0;
    position: relative;
}
```

```
.content-window-category:first-child{
    margin-top: 2rem;
}
.content-window-category:last-child{
    margin-bottom: 2rem;
}
.content-window-category-label{
    width: 100%;
    padding: 5px 10px;
    border-radius: 10px;
    font-size: 1.2rem;
    font-weight: bolder;
}
.content-window-category-items-wrapper{
    display: flex;
    overflow: auto;
    padding: 0 0 0 10px;
}
.content-window-category-item{
    width: var(--content-window-category-item-cover-wrapper-side);
    margin: 1rem 1rem 0.35rem 0;
}
.content-window-category-item-cover-wrapper{
    height: var(--content-window-category-item-cover-wrapper-side);
    width: var(--content-window-category-item-cover-wrapper-side);
    position: relative;
}
.content-window-category-item-cover{
    height: var(--content-window-category-item-cover-wrapper-side);
    width: var(--content-window-category-item-cover-wrapper-side);
    object-fit: cover;
    border-radius: 0.25rem;
    overflow: hidden;
    display: flex;
    flex-wrap: wrap;
}
.content-window-category-item-cover-collage-image{
    height: calc(var(--content-window-category-item-cover-wrapper-side) / 2);
    width: calc(var(--content-window-category-item-cover-wrapper-side) / 2);
    object-fit: cover;
}
.content-window-category-item-cover-overlay{
    height: calc(100%);
    width: 100%;
    position: absolute;
    top: 0;
    left: 0;
```

```
    z-index: 1;
    border-radius: 0.25rem;
    box-shadow: 0 0 50px 50px var(--hover-box-shadow-color) inset;
    cursor: pointer;
    opacity: 0;
}
.content-window-category-item-cover-overlay:hover{
    opacity: 1;
}
.content-window-category-item-cover-overlay-icon-wrapper{
    position: absolute;
    right: 0;
    bottom: 0;
    margin: 0.75rem;
    padding-left: 0.1rem;
    padding-top: 0.05rem;
    box-shadow: 0 0 10px 10px var(--hover-box-shadow-color) inset, 0 0 10px 5px rgba(255, 255, 255, 0.3);
}
.content-window-category-item-cover-overlay-icon-wrapper:hover{
    box-shadow: 0 0 10px 10px black inset, 0 0 10px 10px rgba(255, 255, 255, 0.3);
}
}
.content-window-category-item-labels{
    padding: 10px 5px;
    position: relative;
    height: max-content;
    overflow: hidden;
    font-weight: 500;
}
.content-window-category-item-mainlabel{
    font-size: 0.85rem;
    width: 100%;
    margin-bottom: 5px;
    line-height: 1.2rem;
}
.content-window-category-item-sublabel{
    width: 100%;
    font-size: 0.75rem;
    color: var(--sub-label-color);
    font-weight: 300;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
}
.content-window-category-scroll-button{
    height: 2.5rem;
    width: 2.5rem;
    border-radius: 1.25rem;
```

```
    box-shadow: 0 0 1rem 0.5rem var(--panels-bg-color) inset, 0 0 0.5rem 0 rgba(255, 255, 255, 0.5);
    color: var(--sub-label-color);
    position: absolute;
    top: calc(50% - 1.25rem);
    z-index: 2;
    display: flex;
    justify-content: center;
    align-items: center;
}
.content-window-category-scroll-left-button{
    left: -1.25rem;
}
.content-window-category-scroll-right-button{
    right: -1.25rem;
}
.content-window-category-scroll-button:hover{
    color: white;
    background-color: var(--hover-bg-color-light);
    box-shadow: 0 0 1rem 0.5rem var(--panels-bg-color) inset, 0 0 1rem 0.1rem rgba(255, 255, 255, 0.5);
}


/* PLAYLIST PAGE */

/* playlist description */
#playlist-description{
    width: calc(100% - 4rem);
    background-color: var(--playlist-page-bg-color);
    border-radius: 0.5rem;
    margin-top: 2rem;
    padding: 1.5rem;
    display: flex;
    align-items: center;
    justify-content: start;
}
#playlist-description-cover-wrapper{
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-wrap: wrap;
    margin-right: 2rem;
}
#playlist-description-cover{
    height: var(--playlist-description-cover-side);
    width: var(--playlist-description-cover-side);
    object-fit: cover;
```

```
    border-radius: 0.25rem;
    overflow: hidden;
    display: flex;
    flex-wrap: wrap;
}
.playlist-description-cover-collage-image{
    height: calc(var(--playlist-description-cover-side) / 2);
    width: calc(var(--playlist-description-cover-side) / 2);
    object-fit: cover;
}
#playlist-description-info-and-buttons{
    width: 100%;
    width: calc(100% - var(--playlist-description-cover-side) - 2rem);
}
#playlist-description-info{
    width: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    margin-bottom: 0.5rem;
}
#playlist-description-info > * {
    margin-bottom: 0.5rem;
}
#playlist-info-title{
    font-size: 2rem;
    font-weight: bold;
    margin-bottom: 0.5rem;
}
#playlist-info-artists{
    color: var(--sub-label-color);
    line-height: 1.3rem;
}
#playlist-info-count-and-duration{
    color: var(--sub-label-color);
}
#playlist-description-buttons{
    display: flex;
    flex-wrap: wrap;
    width: 100%;
}
.playlist-description-button{
    height: 1.8rem;
    max-width: fit-content;
    display: block;
    border: 1px solid white;
    border-radius: 2rem;
```

```
    margin-right: 0.5rem;
    margin-bottom: 0.25rem;
    padding: 0.4rem 0.75rem 0.4rem 0.9rem;
    font-size: 0.8rem;
}
.playlist-description-button:hover{
    box-shadow: var(--buttons-box-shadow);
}
.playlist-description-button-icon{
    margin-right: 0.5rem;
}

/* playlist contents */
#playlist-content{
    width: calc(100% - 4rem);
    background-color: var(--playlist-page-bg-color);
    margin: 1rem 0 2rem;
    border-radius: 0.5rem;
    padding: 0 1.5rem 0.75rem;
    display: flex;
    flex-direction: column;
    align-items: center;
    font-size: 0.8rem;
}
#playlist-content hr{
    font-size: 5px;
}
#playlist-content-header{
    width: 100%;
    display: flex;
    align-items: center;
    padding: 0.75rem 0.25rem;
}

#playlist-content-items{
    width: 100%;
}
.playlist-content-item{
    width: 100%;
    display: flex;
    align-items: center;
    padding: 0.25rem;
    border-bottom: 1px solid var(--item-border-bottom-color);
    color: var(--sub-label-color);
}
.playlist-content-item:hover{
    background-color: var(--hover-bg-color-light);
```

```
    color: rgba(255, 255, 255, 0.9);
}

/* playlist contents columns */
.playlist-content-cover-column{
    width: 10%;
    display: inline-flex;
    align-items: center;
    position: relative;
}
.playlist-content-name-column{
    width: 30%;
    padding: 0 0.5rem;
}
#playlist-content .playlist-content-name-column{
    cursor: pointer;
}
.playlist-content-artists-column{
    width: 30%;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    padding: 0 0.5rem;
}
.playlist-content-duration-column{
    width: 15%;
}
.playlist-content-options-column{
    width: 15%;
    display: inline-flex;
}
.playlist-content-options-column button{
    font-size: 0.8rem;
}

/* playlist item cover image */
.playlist-content-item-cover-image{
    height: var(--playlist-content-item-cover-side);
    width: var(--playlist-content-item-cover-side);
    border-radius: 0.2rem;
}
.playlist-content-cover-column .content-item-cover-overlay-icon-wrapper{
    height: var(--playlist-content-item-cover-side);
    width: var(--playlist-content-item-cover-side);
    border-radius: 0.2rem;
    position: absolute;
    top: 0;
```

```
    left: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    z-index: 1;
    box-shadow: 0 0 0.5rem 0.35rem rgba(0, 0, 0, 0.5) inset;
    background-color: rgba(0, 0, 0, 0.5);
    cursor: pointer;
    opacity: 0;
    font-size: 0.7rem;
}
.playlist-content-item:hover .content-item-cover-overlay-icon-wrapper{
    opacity: 1;
}



/* MAIN PLAYER */
.player{
    height: var(--player-height);
    width: 100%;
    min-width: inherit;
    background-color: var(--panels-bg-color);
    box-shadow: var(--panels-box-shadow);
    padding: 5px 10px 15px;
    display: flex;
    align-items: center;
    justify-content: space-around;
    color: white;
    position: fixed;
    bottom: 0;
    left: 0;
    right: 0;
    z-index: 10;
}
.inactive-player{
    bottom: calc(var(--player-height) * -1);
}

/* player left part */
#player-left-part{
    height: 100%;
    width: 25%;
    display: flex;
    align-items: center;
    justify-content: space-evenly;
    margin-top: 0.25rem;
```

```
}
#player-song-cover{
    height: 60px;
    width: 60px;
    border-radius: 0.2rem;
    margin-top: 0.05rem;
}
#player-song-cover-image{
    height: inherit;
    width: inherit;
    border-radius: 0.2rem;
    object-fit: cover;
}
#player-song-info{
    width: calc(90% - 60px);
    max-width: max-content;
    padding: 0.25rem 0.5rem 0 1rem;
}
#player-song-name{
    width: 100%;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
}
#player-song-artist{
    width: 100%;
    font-size: 0.75rem;
    color: var(--sub-label-color);
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
}

/* player middle part */
#player-middle-part{
    height: 100%;
    width: 40%;
    display: flex;
    align-items: center;
}
#player-middle-part-main{
    display: flex;
    flex-direction: column;
    width: 100%;
}
#player-middle-part-controls{
    margin: 0.25rem auto 0;
```

```
    width: 70%;
    display: flex;
    align-items: center;
    justify-content: space-evenly;
}
#repeat-button-container i:first-of-type{
    font-size: 0.45rem;
}
#play-pause-button-container{
    font-size: 1.2rem;
    width: 2.2rem;
    height: 2.2rem;
    border-radius: 1.1rem;
}
#player-seek-bar{
    width: 90%;
    flex-grow: 1;
    margin: auto;
}
#player-middle-part-right{
    height: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;
    position: relative;
}
#player-middle-part-timestamp{
    font-size: 0.75rem;
    color: var(--sub-label-color);
    position: relative;
    top: 4px;
}

/* player right part */
#player-right-part{
    height: 100%;
    width: 30%;
    /* min-width: 300px; */
    display: flex;
    align-items: center;
    justify-content: space-evenly;
    margin-top: 0.25rem;
}
#player-right-part-main-controls{
    height: 100%;
    display: flex;
```

```
    align-items: center;
    justify-content: space-evenly;
}
#player-right-part-volume-controls{
    display: flex;
    align-items: center;
    justify-content: center;
}
#volume-seek-bar{
    width: calc(100% - 2.5rem);
    max-width: 150px;
}

/* player floating part */
#player-floating-part{
    position: absolute;
    right: 1.5rem;
    bottom: 0.7rem;
    height: 0;
    display: flex;
    flex-wrap: nowrap;
    align-items: end;
    z-index: 5;
    color: white;
    margin: 0;
    padding: 0;
}
.player-floating-part-item-overlay{
    height: 100%;
    width: 24%;
    flex-grow: 1;
    min-width: 300px;
    max-width: 350px;
    margin-left: 5px;
    border-radius: 0.25rem;
    box-shadow: var(--overlay-box-shadow);
    background-color: var(--floating-panels-bg-color);
    padding: 0 0.75rem 0.75rem;
    overflow: hidden;
}
.queue-lyrics-overlay-label{
    font-size: 1.1rem;
    color: white;
    padding: 0.5rem 0;
    font-weight: 500;
}
#floating-queue-overlay{
```

```
    margin-right: 0.75rem;
}
.queue-overlay-content-item{
    width: 100%;
    display: flex;
    align-items: center;
    padding: 0.4rem;
    border-bottom: 1px solid var(--item-border-bottom-color);
    color: var(--sub-label-color);
    cursor: pointer;
}
.queue-overlay-content-item:hover{
    background-color: var(--hover-bg-color-light);
    color: rgba(255, 255, 255, 0.9);
}

/* floating queue overlay contents  */
.queue-content-item-cover-column{
    width: 20%;
    display: inline-flex;
    align-items: center;
    position: relative;
}
.queue-content-item-name-and-artists-column{
    width: 65%;
    padding: 0 0.5rem;
    display: inline-flex;
    flex-direction: column;
    align-items: flex-start;
    justify-content: center;
}
.queue-content-item-name-row{
    width: 100%;
    font-size: 0.8rem;
    margin-bottom: 0.1rem;
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
}
.queue-content-item-artists-row{
    width: 100%;
    font-size: 0.7rem;
    font-weight: lighter;
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
}
```

```
.queue-content-item-duration-column{
    width: 15%;
    font-size: 0.7rem;
    font-weight: 300;
}

/* floating queue overlay content item cover image */
.queue-content-item-cover-image{
    height: var(--floating-queue-content-item-cover-side);
    width: var(--floating-queue-content-item-cover-side);
    border-radius: 0.2rem;
}
.queue-content-item-cover-column .content-item-cover-overlay-icon-wrapper{
    height: var(--floating-queue-content-item-cover-side);
    width: var(--floating-queue-content-item-cover-side);
    border-radius: 0.2rem;
    position: absolute;
    top: 0;
    left: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    z-index: 1;
    box-shadow: 0 0 0.5rem 0.35rem rgba(0, 0, 0, 0.5) inset;
    background-color: rgba(0, 0, 0, 0.5);
    cursor: pointer;
    opacity: 0;
}
.queue-overlay-content-item:hover .content-item-cover-overlay-icon-wrapper{
    opacity: 1;
}
#floating-queue-overlay-content-wrapper .content-item-cover-overlay-icon-wrapper{
    font-size: 0.65rem;
}
#floating-lyrics-overlay-content-wrapper{
    font-size: 0.7rem;
    line-height: 1.1rem;
}



/* EXTENDED PLAYER */
.player-extended-overlay{
    position: absolute;
    bottom: 0;
    right: 0;
```

```
    z-index: 10;
    height: calc(100% - var(--header-height));
    width: 100%;
    background-color: var(--panels-bg-color);
    box-shadow: var(--panels-box-shadow);
    padding: 0.75rem 0.5rem 1rem;
    display: flex;
    color: white;
}
.player-extended-overlay-part{
    height: 100%;
    display: inline-block;
    border-radius: 0.25rem;
    margin-right: 0.5rem;
    box-shadow: var(--overlay-box-shadow);
    font-size: 0.8rem;
    overflow: hidden;
    padding: 0 0.5rem;
}
.player-extended-overlay-part:not(:last-child){
    margin-right: 0.5rem;
}
.player-extended-overlay-part-label{
    font-size: 1.2rem;
    color: white;
    padding: 0.5rem 0;
    font-weight: 500;
}

/* extended player queue */
#player-extended-queue{
    width: 30%;
}
#player-extended-queue-content-wrapper .queue-overlay-content-item{
    padding: 0.45rem;
}
#player-extended-queue-content-wrapper .queue-content-item-cover-column{
    min-width: var(--player-extended-queue-content-item-cover-side);
}
.queue-content-item-name-and-artists-column{
    padding: 0 0.5rem;
}
#player-extended-queue-content-wrapper .queue-content-item-name-row{
    font-size: 0.88rem;
    margin-bottom: 0.12rem;
}
#player-extended-queue-content-wrapper .queue-content-item-artists-row{
```

```
    font-size: 0.77rem;
}
#player-extended-queue-content-wrapper .queue-content-item-duration-column{
    font-size: 0.77rem;
}

/* floating queue overlay content item cover image */
#player-extended-queue-content-wrapper .queue-content-item-cover-image{
    height: var(--player-extended-queue-content-item-cover-side);
    width: var(--player-extended-queue-content-item-cover-side);
}
#player-extended-queue-content-wrapper .content-item-cover-overlay-icon-wrapper{
    height: var(--player-extended-queue-content-item-cover-side);
    width: var(--player-extended-queue-content-item-cover-side);
}


#player-extended-cover{
    width: 40%;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 1rem;
}
#player-extended-cover-image{
    object-fit: contain;
    max-height: 100%;
    max-width: 100%;
    border-radius: 0.25rem;
}
#player-extended-lyrics{
    width: 30%;
}
#player-extended-lyrics-content-wrapper{
    font-weight: 300;
    /* padding: 0.5rem; */
    color: var(--sub-label-color);
    line-height: 1.2rem;
}


.queue-lyrics-content-wrapper{
    width: 100%;
    height: calc(100% - 2.5rem);
    overflow-y: auto;
    margin-top: 0.25rem;
}
```

```
.lyrics-content-wrapper{
    font-weight: 300;
    padding: 0.5rem;
    color: var(--sub-label-color);
}



/* FLOATING NOTIFICATION */
.floating-notification{
    position: fixed;
    bottom: calc(var(--player-height) + 0.5rem);
    left: 50%;
    transform: translateX(-50%);
    z-index: 25;
    width: 15rem;
    height: 2.5rem;
    background-color: black;
    box-shadow: 0 0 10px 10px rgba(255, 255, 255, 0.2);
    border-radius: 0.25rem;
    padding: 0.5rem 0.75rem 0.5rem 1rem;
    display: flex;
    align-items: center;
    justify-content: left;
    font-size: 0.8rem;
    color: white;
    transition: all 0.3s ease-in-out;
}
.floating-notification-icon{
    height: 1rem;
    width: 1rem;
    font-size: 1rem;
    margin: 0.5rem 1rem 0.5rem 0;
}



/* GENERAL STYLE CLASSES */
.rotate-180{
    transform: rotate(180deg);
}

.shrunk-element{
    height: 0;
    width: 0;
    min-width: 0;
    max-width: 0;
```

```
    margin: 0;
    padding: 0;
    border: none;
    box-shadow: none;
    overflow: hidden;
}

.invisible-element{
    visibility: hidden;
    opacity: 0;
}

.inactive-button{
    opacity: 0.5;
}

.small-button{
    margin-top: 0.25rem;
    font-size: 0.75rem;
}

.queue-content-item-cover-column .opacity-1{
    opacity: 1;
}

.playlist-content-cover-column .opacity-1{
    opacity: 1;
}

.full-height{
    height: 100%;
}

.active-item{
    background-color: rgba(255, 255, 255, 0.15);
    color: rgba(255, 255, 255, 0.9);
}

.display-none{
    display: none;
}
```

JAVASCRIPT:

```
"use strict";

// Global Variables
const body = document.getElementsByTagName("body")[0];
const body_wallpaper_overlay = document.getElementsByClassName("body-wallpaper-overlay")[0];
const root = document.documentElement;

// Nav
const nav = document.getElementsByTagName("nav")[0];

const nav_buttons = Array.from(document.getElementsByClassName("nav-button-container"));
const nav_home_button_container = document.getElementById("nav-home-button-container");
const nav_explore_button_container = document.getElementById("nav-explore-button-container");
const nav_library_button_container = document.getElementById("nav-library-button-container");
const nav_library_content_items_wrapper = document.getElementsByClassName("nav-library-content-items-wrapper")[0];

const nav_original_width = getComputedStyle(root).getPropertyValue("--nav-width");
const nav_original_min_width = getComputedStyle(root).getPropertyValue("--nav-min-width");
const nav_original_max_width = getComputedStyle(root).getPropertyValue("--nav-max-width");
const nav_shrinking_items = Array.from(document.getElementsByClassName("nav-shrinking-item"));
const nav_button_containers = Array.from(document.getElementsByClassName("nav-button-container"));

// Header
const search_bar = document.getElementById("search-bar");
const search_suggestions_wrapper = document.getElementById("search-suggestions-wrapper");
const profile_button_container = document.getElementById("profile-button-container");
const profile_dropdown_overlay = document.getElementsByClassName("profile-dropdown-overlay")[0];
const profile_dropdown_overlay_login_status = document.getElementById("profile-dropdown-overlay-login-status");
const profile_dropdown_overlay_login_logout_button = document.getElementById("profile-dropdown-overlay-login-logout-button");

// Login Popup
const login_popup = document.getElementsByClassName("login-popup")[0];
const login_popup_fade_bg = document.getElementsByClassName("login-popup-fade-bg")[0];
const login_popup_login_tab_button = document.getElementById("login-popup-login-tab-button");
const login_popup_signup_tab_button = document.getElementById("login-popup-signup-tab-button");
const login_popup_login_tab = document.getElementsByClassName("login-popup-login-tab")[0];
const login_popup_signup_tab = document.getElementsByClassName("login-popup-signup-tab")[0];
const login_popup_close_button = document.getElementById("login-popup-close-button");
const email_regex = /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
```

```
const login_popup_submit_login_button = document.getElementById("login-popup-submit-login-button");
const login_popup_submit_signup_button = document.getElementById("login-popup-submit-signup-button");

// Content Window
const right_window = document.getElementsByClassName("right-window")[0];
const content_window = document.getElementById("content-window");


// Player
const player = document.getElementsByClassName("player")[0];

const player_song_cover_image = document.getElementById("player-song-cover-image");
const player_song_name = document.getElementById("player-song-name");
const player_song_artist = document.getElementById("player-song-artist");

const player_like_button_container = document.getElementById("player-left-part-like-button-container");
const player_like_button_icon = player_like_button_container.firstElementChild;

const player_shuffle_button_container = document.getElementById("shuffle-button-container");
const player_shuffle_button_icon = player_shuffle_button_container.firstElementChild;

const player_previous_song_button_container = document.getElementById("previous-song-button-container");
const player_play_pause_button_container = document.getElementById("play-pause-button-container");
const player_play_pause_button_icon = player_play_pause_button_container.firstElementChild;
const player_next_song_button_container = document.getElementById("next-song-button-container");

const player_repeat_button_container = document.getElementById("repeat-button-container");
const player_repeat_button_icon_one = player_repeat_button_container.firstElementChild;
const player_repeat_button_icon = player_repeat_button_container.lastElementChild;
const player_options_button_container = document.getElementById("player-options-button-container");

const player_audio_controls = document.getElementById("player-audio-controls");
const player_audio_seek_bar = document.getElementById("player-seek-bar");
const player_audio_current_duration_label = document.getElementById("audio-current-duration");
const player_audio_total_duration_label = document.getElementById("audio-total-duration");

const player_queue_button_container = document.getElementById("queue-button-container");
const player_queue_button_icon = player_queue_button_container.firstElementChild;
const player_lyrics_button_container = document.getElementById("lyrics-button-container");
const player_lyrics_button_icon = player_lyrics_button_container.firstElementChild;

const player_volume_seek_bar = document.getElementById("volume-seek-bar");
const player_volume_button_container = document.getElementById("volume-button-container");
const player_volume_button_icon = player_volume_button_container.firstElementChild;

const player_extend_button_container = document.getElementById("player-extend-button-container");
const player_extend_button_icon = player_extend_button_container.firstElementChild;
```

```
const player_floating_part = document.getElementById("player-floating-part");
const player_floating_part_full_height = "calc(100% - var(--header-height) - 12rem)"
const floating_queue_overlay = document.getElementById("floating-queue-overlay");
const floating_queue_overlay_content_wrapper = floating_queue_overlay.getElementsByClassName("queue-lyrics-content-wrapper")[0];
const floating_lyrics_overlay = document.getElementById("floating-lyrics-overlay");
const floating_lyrics_overlay_content_wrapper = floating_lyrics_overlay.getElementsByClassName("queue-lyrics-content-wrapper")[0];

const player_extended_overlay = document.getElementsByClassName("player-extended-overlay")[0];
const player_extended_queue_content_wrapper = document.getElementById("player-extended-queue-content-wrapper");

const player_extended_cover = document.getElementById("player-extended-cover");
player_extended_cover.innerHTML = `<img id="player-extended-cover-image" src="">`;
const player_extended_cover_image = document.getElementById("player-extended-cover-image");

const player_extended_lyrics_content_wrapper = document.getElementById("player-extended-lyrics-content-wrapper");

player_audio_controls.volume = player_volume_seek_bar.value / 100;


// Content Objects
import songs_obj from "./media/obj_jsons/songs.json" assert {type: 'json'};
import songs_root_id_mapping_obj from "./media/obj_jsons/songs_root_id_mapping.json" assert {type: 'json'};
import playlists_obj from "./media/obj_jsons/playlists.json" assert {type: 'json'};
import playlists_root_id_mapping_obj from "./media/obj_jsons/playlists_root_id_mapping.json" assert {type: 'json'};
import tabs_obj from "./media/manual_obj_jsons/tabs.json" assert {type: 'json'};
import categories_obj from "./media/manual_obj_jsons/categories.json" assert {type: 'json'};

var first_play = true;
var current_content_window_content_type = "tab";
var current_content_window_content_tag_id = "home";

var queue_song_id_list = [];
var queue_played_song_indexes_list = [];
var queue_current_song_index;
var shuffle = sessionStorage.getItem("shuffle") ? JSON.parse(sessionStorage.getItem("shuffle")) : false;
var repeat = sessionStorage.getItem("repeat") ? JSON.parse(sessionStorage.getItem("repeat")) : 0;
var current_song_id;

var users_obj = localStorage.getItem("users_obj") ? JSON.parse(localStorage.getItem("users_obj")) : {};
var current_user_id = sessionStorage.getItem("current_user_id") ? JSON.parse(sessionStorage.getItem("current_user_id")) : null;
```

```
var custom_playlists_obj = localStorage.getItem("custom_playlists_obj") ? JSON.parse(localStorage.getItem("custom_playlists_obj")) : {};
var custom_playlist_ids_list = localStorage.getItem("custom_playlist_ids_list") ? JSON.parse(localStorage.getItem("custom_playlist_ids_list")) : [];
var liked_song_ids_list = [];
var saved_playlist_ids_list = [];
var recently_played_song_ids_list = [];
var recently_played_playlist_ids_list = [];


if (shuffle){
    player_shuffle_button_icon.classList.remove("inactive-button");
}
if (repeat == 1){
    player_repeat_button_icon.classList.remove("inactive-button");
}
if (repeat == 2){
    player_repeat_button_icon_one.classList.remove("shrunk-element");
    player_repeat_button_icon.classList.add("small-button");
}


// Functions
// function scrollIntoViewIfNeeded(target) {
//     if (target.getBoundingClientRect().bottom > target.parentElement.innerHeight) {
//         target.scrollIntoView(false);
//     }

//     if (target.getBoundingClientRect().top < 0) {
//         target.scrollIntoView();
//     }
// }

// function scroll_to_item(item) {
//     let item_container = item.getBoundingClientRect();
//     // scroll top of item into view if it is above visible area of content wrapper
//     if (item_container.scrollTop > item.offsetTop) {
//         console.log("scrolling for top");
//         console.log("Parent ScrollTop: " + item_container.scrollTop);
//         console.log("Parent ClientHeight: " + item_container.clientHeight);
//         console.log("Item OffsetTop: " + item.offsetTop);
//         console.log("Item ScrollHeight: " + item.scrollHeight);
//         console.log(" ");
//         item.scrollIntoView();
//     }
//     // scroll bottom of item into view if it is under visible area of content wrapper
//     if (item_container.scrollTop + item_container.clientHeight < item.offsetTop + item.scrollHeight) {
```

```
//         console.log("scrolling for bottom");
//         console.log("Parent ScrollTop: " + item_container.scrollTop);
//         console.log("Parent ClientHeight: " + item_container.clientHeight);
//         console.log("Item OffsetTop: " + item.offsetTop);
//         console.log("Item ScrollHeight: " + item.scrollHeight);
//         console.log(" ");
//         item.scrollIntoView(false);
//     }
// }

function shuffle_array(array) {
    let returnArray = [...array], currentIndex = returnArray.length,  randomIndex;
    // While there remain elements to shuffle.
    while (currentIndex != 0) {
      // Pick a remaining element.
      randomIndex = Math.floor(Math.random() * currentIndex);
      currentIndex--;
      // And swap it with the current element.
      [returnArray[currentIndex], returnArray[randomIndex]] = [returnArray[randomIndex],
returnArray[currentIndex]];
    }
    return returnArray;
}

function return_duration_string(duration){
    return new Date(duration * 1000).toISOString().slice(14, 19);
}

function update_audio_seek_bar_and_time_stamp(){
    // updating seek bar current position and max value
    player_audio_seek_bar.value = player_audio_controls.currentTime;
    player_audio_seek_bar.max = player_audio_controls.duration;
    // updating current timestamp
    player_audio_current_duration_label.innerText = return_duration_string(player_audio_controls.currentTime);
}

function render_nav_library_content(){
    // clearing nav library content
    nav_library_content_items_wrapper.innerHTML =
        `<p class="nav-library-content-item">History</p>
        <p class="nav-library-content-item">Liked Songs</p>`;

    // creating nav library content
    for (let playlist_index = 0; playlist_index < Object.keys(playlists_obj).length; playlist_index++){
        let playlist_id = Number(Object.keys(playlists_obj)[playlist_index]);
        let playlist_name = playlists_obj[playlist_id].playlist_name;
```

```
        let nav_library_content_item = document.createElement("p");
        nav_library_content_item.className = "nav-library-content-item";
        nav_library_content_item.innerText = playlist_name;
        nav_library_content_items_wrapper.appendChild(nav_library_content_item);
        nav_library_content_item.onclick = (event) => {
            event.stopPropagation();
            render_playlist(playlist_id);
        }
    }
}

function show_hide_search_suggestions(show = true){
    if (show){
        search_suggestions_wrapper.classList.remove("shrunk-element");
    }
    else{
        search_suggestions_wrapper.classList.add("shrunk-element");
    }
}

function update_search_suggestions(){
    search_suggestions_wrapper.innerHTML = "";

    let search_query = search_bar.value.toLowerCase();

    for (let song_id in songs_obj){

        if (search_suggestions_wrapper.childElementCount >= 5) return;

        let song_obj = songs_obj[song_id];

        let query_in_song_name = song_obj.song_name.toLowerCase().includes(search_query);
        let query_in_artist_names = song_obj.song_artist_names.filter(
            artist_name => artist_name.split(" ").filter(
                word => word.toLowerCase().startsWith(search_query)
                ).length
            ).length;

        if (query_in_song_name || query_in_artist_names){

            let search_suggestion = document.createElement("span");
            search_suggestions_wrapper.appendChild(search_suggestion);
            search_suggestion.className = "search-suggestion";

            let search_suggestion_text = document.createElement("p");
            search_suggestion_text.className = "search-suggestion-text";
            search_suggestion_text.innerHTML = song_obj.song_artist_names + " - " + song_obj.song_name;
```

```
            search_suggestion.appendChild(search_suggestion_text);

            let search_icon = document.createElement("i");
            search_icon.classList = "fa-solid fa-magnifying-glass search-bar-icon";
            search_suggestion.append(search_icon);

            search_suggestion.onclick = (event) => {
                event.stopPropagation();
                play_song(song_id);
                show_hide_search_suggestions(false);
            }
        }
    }
    while (search_suggestions_wrapper.childElementCount < 5){

        let search_suggestion = document.createElement("span");
        search_suggestions_wrapper.appendChild(search_suggestion);
        search_suggestion.className = "search-suggestion";

        let search_suggestion_text = document.createElement("p");
        search_suggestion_text.className = "search-suggestion-text";
        search_suggestion_text.innerHTML = `Search History ${search_suggestions_wrapper.childElementCount}`;
        search_suggestion.appendChild(search_suggestion_text);

        let search_icon = document.createElement("i");
        search_icon.classList = "fa-solid fa-magnifying-glass search-bar-icon";
        search_suggestion.append(search_icon);

        search_suggestion.onclick = (event) => {
            event.stopPropagation();
            play_song(song_id);
        }
    }
}

function render_content_window(tab_tag, only_update_content = false){
    current_content_window_content_type = "tab";
    current_content_window_content_tag_id = tab_tag;

    if (! only_update_content){
        toggle_extended_player(false);      // to close extended player when tab clicked with it open
        content_window.scrollTo(0, 0);
    }

    content_window.innerHTML = "";

    let tab_obj = tabs_obj[tab_tag];
```

```
    let tab_category_ids = tab_obj.tab_category_ids;

    if (! current_user_id && tab_tag == "library") {
        tab_category_ids = Object.keys(categories_obj).slice(12, 18);
    }
    tab_category_ids = tab_category_ids.concat(Object.keys(categories_obj).slice(12, 18-
tab_category_ids.length).map(category_id => Number(category_id)));

    for (let category_index in tab_category_ids){

        let category_id = tab_category_ids[category_index];

        // getting category object
        let category_obj = categories_obj[category_id];

        let category_tag = category_obj.category_tag;
        let category_name = category_obj.category_name;
        let category_items_type = category_obj.category_items_type;
        let category_item_ids = category_obj.category_item_ids;

        if (tab_tag == "library" && category_tag == "recents") category_name = "History";

        if (!current_user_id && ["recents", "liked", "saved_playlists"].includes(category_tag)) continue;

        if (category_tag == "recents"){
            if (recently_played_song_ids_list.length == 0) continue;
            category_item_ids = recently_played_song_ids_list;

        }
        if (category_tag == "liked"){
            if (liked_song_ids_list.length == 0) continue;
            category_item_ids = liked_song_ids_list;

        }
        if (category_tag == "saved_playlists"){
            if (saved_playlist_ids_list.length == 0) continue;
            category_item_ids = saved_playlist_ids_list;

        }

        // creating content window category
        let content_window_category = document.createElement("div");
        content_window_category.id = category_tag + "-category";
        content_window_category.classList.add("content-window-category");
        content_window.appendChild(content_window_category);
```

```
        // creating scroll buttons
        let content_window_category_scroll_left_button = document.createElement("button");
        content_window_category_scroll_left_button.classList.add("content-window-category-scroll-button");
        content_window_category_scroll_left_button.classList.add("content-window-category-scroll-left-button");
        content_window_category_scroll_left_button.classList.add("invisible-element");
        content_window_category_scroll_left_button.innerHTML = "<i class='fa-solid fa-chevron-left'></i>";
        content_window_category.appendChild(content_window_category_scroll_left_button);

        let content_window_category_scroll_right_button = document.createElement("button");
        content_window_category_scroll_right_button.classList.add("content-window-category-scroll-button");
        content_window_category_scroll_right_button.classList.add("content-window-category-scroll-right-
button");
        content_window_category_scroll_right_button.innerHTML = "<i class='fa-solid fa-chevron-right'></i>";
        content_window_category.appendChild(content_window_category_scroll_right_button);

        // creating category label and items wrapper
        let content_window_category_label = document.createElement("p");
        content_window_category_label.classList.add("content-window-category-label");
        content_window_category_label.innerText = category_name;
        content_window_category.appendChild(content_window_category_label);

        let content_window_category_items_wrapper = document.createElement("div");
        content_window_category_items_wrapper.classList.add("content-window-category-items-wrapper");
        content_window_category.appendChild(content_window_category_items_wrapper);

        // adding scroll buttons event listeners
        content_window_category_scroll_left_button.onclick = () => {
            let scroll_amount = (Math.floor(content_window_category_items_wrapper.clientWidth/200) - 1) * 200;
            content_window_category_items_wrapper.scrollLeft -= scroll_amount;
        };
        content_window_category_scroll_right_button.onclick = () => {
            let scroll_amount = (Math.floor(content_window_category_items_wrapper.clientWidth/200) - 1) * 200;
            content_window_category_items_wrapper.scrollLeft += scroll_amount;
        };

        // adding scroll event listener to category items wrapper to show/hide scroll buttons
        content_window_category_items_wrapper.onscroll = () => {
            if (content_window_category_items_wrapper.scrollLeft == 0){
                content_window_category_scroll_left_button.classList.add("invisible-element");
            }
            else{
                content_window_category_scroll_left_button.classList.remove("invisible-element");
            }
            if (content_window_category_items_wrapper.scrollLeft >=
content_window_category_items_wrapper.scrollWidth -
content_window_category_items_wrapper.clientWidth){
                content_window_category_scroll_right_button.classList.add("invisible-element");
```

```
            }
            else{
                content_window_category_scroll_right_button.classList.remove("invisible-element");
            }
        }

        // adding items to category items wrapper
        for (let category_item_index in category_item_ids){

            let category_item_id = category_item_ids[category_item_index];

            let content_window_category_item = document.createElement("div");
            content_window_category_item.classList.add("content-window-category-item");
            content_window_category_items_wrapper.appendChild(content_window_category_item);

            let content_window_category_item_cover_wrapper = document.createElement("div");
            content_window_category_item_cover_wrapper.classList.add("content-window-category-item-cover-
wrapper");
            content_window_category_item.appendChild(content_window_category_item_cover_wrapper);

            let category_item_obj, content_window_category_mainlabel, content_window_category_sublabel;

            let content_window_category_item_cover;

            if (category_items_type == "song"){
                category_item_obj = songs_obj[category_item_id];
                content_window_category_mainlabel = category_item_obj.song_name;
                content_window_category_sublabel = category_item_obj.song_artist_names;

                // adding image to cover if category item is a song
                content_window_category_item_cover = document.createElement("img");
                content_window_category_item_cover.src = "./media/covers/" +
songs_obj[category_item_id].song_file_name_root + ".jpg";
                content_window_category_item_cover.alt = category_tag + "_item_cover";
            }
            else{
                category_item_obj = playlists_obj[category_item_id];
                content_window_category_mainlabel = category_item_obj.playlist_name;
                content_window_category_sublabel = category_item_obj.playlist_song_artist_names.join(", ");

                // adding collage container div to cover if category item is a playlist and has 4 or more songs
                if (category_item_obj.playlist_song_ids.length >= 4){
                    content_window_category_item_cover = document.createElement("div");
                    for (let i = 0; i < 4; i++){
                        let content_window_category_item_cover_collage_image = document.createElement("img");
                        content_window_category_item_cover_collage_image.src = "./media/covers/" +
songs_obj[category_item_obj.playlist_song_ids[i]].song_file_name_root + ".jpg";
```

```
                content_window_category_item_cover_collage_image.classList.add("content-window-category-item-cover-collage-image");
                content_window_category_item_cover_collage_image.alt = category_tag + "_item_cover";

content_window_category_item_cover.appendChild(content_window_category_item_cover_collage_image);
            }
        }
        // otherwise adding single image to cover
        else{
            content_window_category_item_cover = document.createElement("img");
            content_window_category_item_cover.src = "./media/covers/" + songs_obj[category_item_obj.playlist_song_ids[0]].song_file_name_root + ".jpg";
            content_window_category_item_cover.alt = category_tag + "_item_cover";
        }
    }

    content_window_category_item_cover.classList.add("content-window-category-item-cover");
    content_window_category_item_cover_wrapper.appendChild(content_window_category_item_cover);

    let content_window_category_item_cover_overlay = document.createElement("div");
    content_window_category_item_cover_overlay.classList.add("content-window-category-item-cover-overlay");

content_window_category_item_cover_wrapper.appendChild(content_window_category_item_cover_overlay);

    let content_window_category_item_cover_overlay_icon_wrapper = document.createElement("div");
    content_window_category_item_cover_overlay_icon_wrapper.classList.add("standard-button");
    content_window_category_item_cover_overlay_icon_wrapper.classList.add("content-window-category-item-cover-overlay-icon-wrapper");

content_window_category_item_cover_overlay.appendChild(content_window_category_item_cover_overlay_icon_wrapper);

    let content_window_category_overlay_wrapper_icon = document.createElement("i");
    content_window_category_overlay_wrapper_icon.classList.add("fa-solid", "fa-play");

content_window_category_item_cover_overlay_icon_wrapper.appendChild(content_window_category_overlay_wrapper_icon);

    let content_window_category_item_labels = document.createElement("div");
    content_window_category_item_labels.classList.add("content-window-category-item-labels");
    content_window_category_item.appendChild(content_window_category_item_labels);

    let content_window_category_item_mainlabel = document.createElement("p");
    content_window_category_item_mainlabel.classList.add("content-window-category-item-mainlabel");
    content_window_category_item_mainlabel.innerText = content_window_category_mainlabel;
    content_window_category_item_labels.appendChild(content_window_category_item_mainlabel);
```

```
            let content_window_category_item_sublabel = document.createElement("p");
            content_window_category_item_sublabel.classList.add("content-window-category-item-sublabel");
            content_window_category_item_sublabel.innerText = content_window_category_sublabel;
            content_window_category_item_labels.appendChild(content_window_category_item_sublabel);

            // add event listener to play that song or render and play that playlist
            if (category_items_type == "song"){
                content_window_category_item.onclick = () => play_song(category_item_id);
            }
            else{
                content_window_category_item.onclick = () => render_playlist(category_item_id);
                content_window_category_item_cover_overlay_icon_wrapper.onclick = () => {
                    play_song(category_item_obj.playlist_song_ids[0], category_item_id);
                }
            }
        }
    }

    for (let nav_button_index in nav_buttons){
        let nav_button = nav_buttons[nav_button_index];
        nav_button.classList.remove("active-item");
        if (nav_button.id == `nav-${tab_tag}-button-container`){
            nav_button.classList.add("active-item");
        }
    }
}

function render_playlist(playlist_id){
    current_content_window_content_type = "playlist";
    current_content_window_content_tag_id = playlist_id;

    playlist_id = Number(playlist_id);
    content_window.innerHTML = "";

    let playlist_obj = playlists_obj[playlist_id];

    let playlist_name = playlist_obj.playlist_name;
    let playlist_song_artist_names = playlist_obj.playlist_song_artist_names.join(", ");
    let playlist_song_ids = playlist_obj.playlist_song_ids;

    // playlist description
    let playlist_description = document.createElement("div");
    playlist_description.id = "playlist-description";
    content_window.appendChild(playlist_description);
```

```
    // playlist descripton cover
    let playlist_description_cover_wrapper = document.createElement("div");
    playlist_description_cover_wrapper.id = "playlist-description-cover-wrapper";
    playlist_description.appendChild(playlist_description_cover_wrapper);

    let playlist_description_cover;

    // adding collage container div to cover if category item is a playlist and has 4 or more songs
    if (playlist_song_ids.length >= 4){
        playlist_description_cover = document.createElement("div");
        for (let i = 0; i < 4; i++){
            let playlist_description_cover_collage_image = document.createElement("img");
            playlist_description_cover_collage_image.src = "./media/covers/" +
songs_obj[playlist_song_ids[i]].song_file_name_root + ".jpg";
            playlist_description_cover_collage_image.classList.add("playlist-description-cover-collage-image");
            playlist_description_cover_collage_image.alt = "playlist_cover";
            playlist_description_cover.appendChild(playlist_description_cover_collage_image);
        }
    }
    // otherwise adding single image to cover
    else{
        playlist_description_cover = document.createElement("img");
        playlist_description_cover.src = "./media/covers/" + songs_obj[playlist_song_ids[0]].song_file_name_root +
".jpg";
        playlist_description_cover.alt = "playlist_cover";
    }

    playlist_description_cover.id = "playlist-description-cover";
    playlist_description_cover_wrapper.appendChild(playlist_description_cover);

    // playlist description info and buttons
    let playlist_description_info_and_buttons = document.createElement("div");
    playlist_description_info_and_buttons.id = "playlist-description-info-and-buttons";
    playlist_description.appendChild(playlist_description_info_and_buttons);

    // playlist description info
    let playlist_description_info = document.createElement("div");
    playlist_description_info.id = "playlist-description-info";
    playlist_description_info_and_buttons.appendChild(playlist_description_info);

    let playlist_info_title = document.createElement("p");
    playlist_info_title.id = "playlist-info-title";
    playlist_info_title.innerText = playlist_name;
    playlist_description_info.appendChild(playlist_info_title);

    let playlist_info_artists = document.createElement("p");
    playlist_info_artists.id = "playlist-info-artists";
```

```
playlist_info_artists.innerText = playlist_song_artist_names;
playlist_description_info.appendChild(playlist_info_artists);

let playlist_info_count_and_duration = document.createElement("div");
playlist_info_count_and_duration.id = "playlist-info-count-and-duration";
playlist_description_info.appendChild(playlist_info_count_and_duration);

let playlist_info_song_count = document.createElement("p");
playlist_info_song_count.id = "playlist-info-song-count";
playlist_info_count_and_duration.appendChild(playlist_info_song_count);

let playlist_info_song_count_number = document.createElement("span");
playlist_info_song_count_number.id = "playlist-info-song-count-number";
playlist_info_song_count_number.innerText = playlist_song_ids.length;
playlist_info_song_count.appendChild(playlist_info_song_count_number);

let playlist_info_song_count_label = document.createElement("span");
playlist_info_song_count_label.id = "playlist-info-song-count-label";
playlist_info_song_count_label.innerHTML = " songs  •  ";
playlist_info_song_count.appendChild(playlist_info_song_count_label);

let playlist_info_duration = document.createElement("p");
playlist_info_duration.id = "playlist-info-duration";
playlist_info_count_and_duration.appendChild(playlist_info_duration);

let playlist_info_duration_sum = 0;
let playlist_info_duration_number = document.createElement("span");
playlist_info_duration_number.id = "playlist-info-duration-number";
playlist_info_duration.appendChild(playlist_info_duration_number);

let playlist_info_duration_label = document.createElement("span");
playlist_info_duration_label.id = "playlist-info-duration-label";
playlist_info_duration_label.innerHTML = " minutes";
playlist_info_duration.appendChild(playlist_info_duration_label);


// playlist description buttons
let playlist_description_buttons = document.createElement("div");
playlist_description_buttons.id = "playlist-description-buttons";
playlist_description_info_and_buttons.appendChild(playlist_description_buttons);

// play button
let playlist_description_play_button = document.createElement("button");
playlist_description_play_button.id = "playlist-description-play-button";
playlist_description_play_button.classList.add("playlist-description-button");
playlist_description_play_button.title = "Play";
playlist_description_buttons.appendChild(playlist_description_play_button);
```

```
let playlist_description_play_button_icon = document.createElement("i");
playlist_description_play_button_icon.classList.add("fa-solid");
playlist_description_play_button_icon.classList.add("fa-play");
playlist_description_play_button_icon.classList.add("playlist-description-button-icon");
playlist_description_play_button.appendChild(playlist_description_play_button_icon);

let playlist_description_play_button_label = document.createElement("p");
playlist_description_play_button_label.classList.add("playlist-description-button-label");
playlist_description_play_button_label.innerText = "Play";
playlist_description_play_button.appendChild(playlist_description_play_button_label);

// shuffle play button
let playlist_description_shuffle_play_button = document.createElement("button");
playlist_description_shuffle_play_button.id = "playlist-description-shuffle-play-button";
playlist_description_shuffle_play_button.classList.add("playlist-description-button");
playlist_description_shuffle_play_button.title = "Shuffle Play";
playlist_description_buttons.appendChild(playlist_description_shuffle_play_button);

let playlist_description_shuffle_play_button_icon = document.createElement("i");
playlist_description_shuffle_play_button_icon.classList.add("fa-solid");
playlist_description_shuffle_play_button_icon.classList.add("fa-shuffle");
playlist_description_shuffle_play_button_icon.classList.add("playlist-description-button-icon");
playlist_description_shuffle_play_button.appendChild(playlist_description_shuffle_play_button_icon);

let playlist_description_shuffle_play_button_label = document.createElement("p");
playlist_description_shuffle_play_button_label.classList.add("playlist-description-button-label");
playlist_description_shuffle_play_button_label.innerText = "Shuffle Play";
playlist_description_shuffle_play_button.appendChild(playlist_description_shuffle_play_button_label);

// add to library button
let playlist_description_add_to_library_button = document.createElement("button");
playlist_description_add_to_library_button.id = "playlist-description-add-to-library-button";
playlist_description_add_to_library_button.classList.add("playlist-description-button");
playlist_description_add_to_library_button.title = "Add to Library";
playlist_description_buttons.appendChild(playlist_description_add_to_library_button);

let playlist_description_add_to_library_button_icon = document.createElement("i");
playlist_description_add_to_library_button_icon.classList.add("fa-solid");
playlist_description_add_to_library_button_icon.classList.add("fa-plus");
playlist_description_add_to_library_button_icon.classList.add("playlist-description-button-icon");
playlist_description_add_to_library_button.appendChild(playlist_description_add_to_library_button_icon);

let playlist_description_add_to_library_button_label = document.createElement("p");
playlist_description_add_to_library_button_label.classList.add("playlist-description-button-label");
playlist_description_add_to_library_button_label.innerText = "Add to Library";
playlist_description_add_to_library_button.appendChild(playlist_description_add_to_library_button_label);
```

```
    if (saved_playlist_ids_list.includes(playlist_id)) {
        playlist_description_add_to_library_button.title = "Remove from Library";
        playlist_description_add_to_library_button_icon.classList.remove("fa-plus");
        playlist_description_add_to_library_button_icon.classList.add("fa-square-minus");
        playlist_description_add_to_library_button_label.innerText = "Remove from Library";
    }

    // more options button
    let playlist_description_add_to_queue_button = document.createElement("button");
    playlist_description_add_to_queue_button.id = "playlist-description-play-button";
    playlist_description_add_to_queue_button.classList.add("playlist-description-button");
    playlist_description_add_to_queue_button.title = "Add to Queue";
    playlist_description_buttons.appendChild(playlist_description_add_to_queue_button);

    let playlist_description_add_to_queue_button_icon = document.createElement("i");
    playlist_description_add_to_queue_button_icon.classList.add("fa-solid");
    playlist_description_add_to_queue_button_icon.classList.add("fa-list-check");
    playlist_description_add_to_queue_button_icon.classList.add("playlist-description-button-icon");
    playlist_description_add_to_queue_button.appendChild(playlist_description_add_to_queue_button_icon);

    let playlist_description_add_to_queue_button_label = document.createElement("p");
    playlist_description_add_to_queue_button_label.classList.add("playlist-description-button-label");
    playlist_description_add_to_queue_button_label.innerText = "Add to Queue";
    playlist_description_add_to_queue_button.appendChild(playlist_description_add_to_queue_button_label);

    // playlist buttons event listeners
    playlist_description_play_button.onclick = () => play_song(playlist_song_ids[0], playlist_id);

    playlist_description_shuffle_play_button.onclick = () => play_song(playlist_song_ids[Math.floor(Math.random() * playlist_song_ids.length)], playlist_id, false, true);

    playlist_description_add_to_library_button.onclick = () => add_remove_playlist_to_library(playlist_id);

    playlist_description_add_to_queue_button.onclick = () => {
        queue_song_id_list = queue_song_id_list.concat(playlist_song_ids);
        render_queue();
    };


    // playlist contents
    let playlist_content = document.createElement("div");
    playlist_content.id = "playlist-content";
    content_window.appendChild(playlist_content);

    // playlist contents header
    let playlist_content_header = document.createElement("div");
```

```
playlist_content_header.id = "playlist-content-header";
playlist_content.appendChild(playlist_content_header);

let playlist_content_header_cover_space = document.createElement("div");
playlist_content_header_cover_space.classList.add("playlist-content-cover-column");
playlist_content_header.appendChild(playlist_content_header_cover_space);

let playlist_content_header_name = document.createElement("p");
playlist_content_header_name.classList.add("playlist-content-name-column");
playlist_content_header_name.innerText = "Name";
playlist_content_header.appendChild(playlist_content_header_name);

let playlist_content_header_artists = document.createElement("p");
playlist_content_header_artists.classList.add("playlist-content-artists-column");
playlist_content_header_artists.innerText = "Artists";
playlist_content_header.appendChild(playlist_content_header_artists);

let playlist_content_header_duration = document.createElement("p");
playlist_content_header_duration.classList.add("playlist-content-duration-column");
playlist_content_header_duration.innerText = "Duration";
playlist_content_header.appendChild(playlist_content_header_duration);

let playlist_content_header_options = document.createElement("p");
playlist_content_header_options.classList.add("playlist-content-options-column");
playlist_content_header_options.innerText = "Options";
playlist_content_header.appendChild(playlist_content_header_options);

let header_hr_line = document.createElement("hr");
playlist_content.appendChild(header_hr_line);

// playlist contents body
let playlist_content_items = document.createElement("div");
playlist_content_items.id = "playlist-content-items";
playlist_content.appendChild(playlist_content_items);

// playlist contents items
for (let song_index in playlist_song_ids){

    // getting song object and info
    let song_id = Number(playlist_song_ids[song_index]);
    let song_obj = songs_obj[song_id];

    let song_name = song_obj.song_name;
    let song_artist_names = song_obj.song_artist_names;
    let song_duration = song_obj.song_duration;

    // playlist item
```

```
        let playlist_content_item = document.createElement("div");
        playlist_content_item.classList.add("playlist-content-item");
        playlist_content_item.id = "playlist-content-item-song-" + song_id;
        playlist_content_items.appendChild(playlist_content_item);

        // playlist item cover
        let playlist_content_item_cover = document.createElement("div");
        playlist_content_item_cover.classList.add("playlist-content-cover-column");
        playlist_content_item_cover.title = "Play";
        playlist_content_item.appendChild(playlist_content_item_cover);

        let playlist_content_item_cover_image = document.createElement("img");
        playlist_content_item_cover_image.classList.add("playlist-content-item-cover-image");
        playlist_content_item_cover_image.src = "./media/covers/" + songs_obj[song_id].song_file_name_root +
".jpg";
        playlist_content_item_cover_image.alt = "playlist_item_cover";
        playlist_content_item_cover.appendChild(playlist_content_item_cover_image);

        let playlist_content_item_cover_image_overlay_icon_wrapper = document.createElement("div");
        playlist_content_item_cover_image_overlay_icon_wrapper.classList.add("content-item-cover-overlay-icon-
wrapper");
        playlist_content_item_cover.appendChild(playlist_content_item_cover_image_overlay_icon_wrapper);

        let playlist_content_item_cover_image_overlay_icon = document.createElement("i");
        playlist_content_item_cover_image_overlay_icon.classList.add("fa-solid");
        playlist_content_item_cover_image_overlay_icon.classList.add("fa-play");
        playlist_content_item_cover_image_overlay_icon.classList.add("playlist-content-item-cover-image-
overlay-icon");

playlist_content_item_cover_image_overlay_icon_wrapper.appendChild(playlist_content_item_cover_image_ov
erlay_icon);

        // playlist item name
        let playlist_content_item_name = document.createElement("p");
        playlist_content_item_name.classList.add("playlist-content-name-column");
        playlist_content_item_name.innerText = song_name;
        playlist_content_item_name.title = song_name;
        playlist_content_item.appendChild(playlist_content_item_name);

        // playlist item artists
        let playlist_content_item_artists = document.createElement("p");
        playlist_content_item_artists.classList.add("playlist-content-artists-column");
        playlist_content_item_artists.innerText = song_artist_names;
        playlist_content_item_artists.title = song_artist_names;
        playlist_content_item.appendChild(playlist_content_item_artists);

        // playlist item duration
```

```
        let playlist_content_item_duration = document.createElement("p");
        playlist_content_item_duration.classList.add("playlist-content-duration-column");
        playlist_content_item_duration.innerText = return_duration_string(song_duration);
        playlist_info_duration_sum += song_duration;
        playlist_content_item.appendChild(playlist_content_item_duration);


        // playlist item options
        let playlist_content_item_options = document.createElement("div");
        playlist_content_item_options.classList.add("playlist-content-options-column");
        playlist_content_item.appendChild(playlist_content_item_options);

        // playlist item option like button
        let playlist_content_item_like_button = document.createElement("button");
        playlist_content_item_like_button.classList.add("playlist-content-item-like-button");
        playlist_content_item_like_button.classList.add("standard-button");
        playlist_content_item_options.appendChild(playlist_content_item_like_button);

        let playlist_content_item_like_button_icon = document.createElement("i");
        if (liked_song_ids_list.includes(song_id)){
            playlist_content_item_like_button_icon.classList.add("fa-solid");
        } else {
            playlist_content_item_like_button_icon.classList.add("fa-regular");
        }
        playlist_content_item_like_button_icon.classList.add("fa-heart");
        playlist_content_item_like_button_icon.id = "playlist-content-item-like-button-icon-song-" + song_id;
        playlist_content_item_like_button.appendChild(playlist_content_item_like_button_icon);

        playlist_content_item_like_button.onclick = () => {
            like_unlike_song(song_id);
        };

        // playlist item option options button
        let playlist_content_item_options_button = document.createElement("button");
        playlist_content_item_options_button.classList.add("standard-button");
        playlist_content_item_options.appendChild(playlist_content_item_options_button);

        let playlist_content_item_options_button_icon = document.createElement("i");
        playlist_content_item_options_button_icon.classList.add("fa-solid");
        playlist_content_item_options_button_icon.classList.add("fa-ellipsis-vertical");
        playlist_content_item_options_button.appendChild(playlist_content_item_options_button_icon);

        // highlighting current playing song in playlist contents
        // highlight_playing_item alone doesn't do it because category_item_overlay_icon_wrapper event calls play_song before the category_item_overlay event that renders playlist
        // so the play_song function is called before the playlist is rendered and the current song is not highlighted
        if (song_id == current_song_id){
```

```
            playlist_content_item.classList.add("active-item");
            playlist_content_item.classList.add("active-item");
            playlist_content_item_cover_image_overlay_icon.classList.remove("fa-play");
            playlist_content_item_cover_image_overlay_icon.classList.add("fa-volume-high");
            playlist_content_item_cover_image_overlay_icon_wrapper.classList.add("opacity-1");
        }

        // add event listeners to play that song
        playlist_content_item_cover.addEventListener("click", function(){
            play_song(song_id);
        });
        playlist_content_item_name.addEventListener("click", function(){
            play_song(song_id);
        });
    }

    playlist_info_duration_number.innerText = Math.ceil(playlist_info_duration_sum / 60);

    content_window.scrollTo(0, 0);

    let nav_last_active_tab = nav.querySelector(".active-item");
    if (nav_last_active_tab) nav_last_active_tab.classList.remove("active-item");
}

function add_remove_playlist_to_library(playlist_id){
    if (!current_user_id){
        make_floating_notification("not_logged_in", false);
        return;
    }
    playlist_id = Number(playlist_id);
    if (saved_playlist_ids_list.includes(playlist_id)){
        saved_playlist_ids_list = saved_playlist_ids_list.filter(id => id != playlist_id);
        make_floating_notification("remove_playlist_from_library");
    } else {
        saved_playlist_ids_list.unshift(playlist_id);
        make_floating_notification("add_playlist_to_library");
    }
    update_user_content();
}

function update_queue(item_id, is_song = false, append_queue = false, shuffle_current_queue = false){
    item_id = Number(item_id);
    // is_song == true means item_id is a song id
    // replacing queue with the new song's playlist
    if (is_song) {
        if ( ! queue_song_id_list.includes(item_id) ){
            let playlist_id = songs_obj[item_id].song_playlist_ids[0];
```

```
            queue_song_id_list = [...playlists_obj[playlist_id].playlist_song_ids].map(song_id => Number(song_id));
            queue_song_id_list = queue_song_id_list.filter(song_id => song_id != current_song_id);
            queue_song_id_list.unshift(current_song_id);
        }
    }
    // appending queue with new playlist's songs if append_queue == true
    // otherwise replacing queue with new playlist's songs if append_queue == false
    else {
        if (append_queue){
            make_floating_notification("add_playlist_to_queue");
            let appending_song_ids = [...playlists_obj[item_id].playlist_song_ids].map(song_id => Number(song_id));
            if (shuffle_current_queue){
                appending_song_ids = shuffle_array(appending_song_ids);
            }
            // removing current song and songs already in queue from appending_song_ids
            appending_song_ids = appending_song_ids.filter(song_id => song_id != current_song_id && ! queue_song_id_list.includes(song_id));
            if (!first_play) appending_song_ids.unshift(current_song_id);
            queue_song_id_list = queue_song_id_list.concat(appending_song_ids);
        } else {
            if (shuffle_current_queue){
                queue_song_id_list = shuffle_array([...playlists_obj[item_id].playlist_song_ids].map(song_id => Number(song_id)));
                queue_song_id_list = queue_song_id_list.filter(song_id => song_id != current_song_id);
                queue_song_id_list.unshift(current_song_id);
            }
            else{
                queue_song_id_list = [...playlists_obj[item_id].playlist_song_ids].map(song_id => Number(song_id));
            }
        }
    }

    queue_current_song_index = queue_song_id_list.lastIndexOf(current_song_id);

    render_queue();
}

function render_queue(){
    floating_queue_overlay_content_wrapper.innerHTML = "";
    player_extended_queue_content_wrapper.innerHTML = "";

    for (let song_index in queue_song_id_list){

        // getting song object and info
        let song_id = queue_song_id_list[song_index];
        let song_obj = songs_obj[song_id];
```

```
        let song_name = song_obj.song_name;
        let song_artist_names = song_obj.song_artist_names;
        let song_duration = song_obj.song_duration;

        // creating queue overlay item
        let queue_overlay_content_item = document.createElement("div");
        queue_overlay_content_item.classList.add("queue-overlay-content-item");
        queue_overlay_content_item.id = "queue-overlay-content-item-song-" + song_id;
        floating_queue_overlay_content_wrapper.appendChild(queue_overlay_content_item);

        // queue overlay item cover
        let queue_overlay_content_item_cover = document.createElement("div");
        queue_overlay_content_item_cover.classList.add("queue-content-item-cover-column");
        queue_overlay_content_item.appendChild(queue_overlay_content_item_cover);

        let queue_overlay_content_item_cover_image = document.createElement("img");
        queue_overlay_content_item_cover_image.classList.add("queue-content-item-cover-image");
        queue_overlay_content_item_cover_image.src = "./media/covers/" +
songs_obj[song_id].song_file_name_root + ".jpg";
        queue_overlay_content_item_cover_image.alt = "queue_item_cover";
        queue_overlay_content_item_cover.appendChild(queue_overlay_content_item_cover_image);

        let queue_overlay_content_item_cover_image_overlay_icon_wrapper = document.createElement("div");
        queue_overlay_content_item_cover_image_overlay_icon_wrapper.classList.add("content-item-cover-
overlay-icon-wrapper");

queue_overlay_content_item_cover.appendChild(queue_overlay_content_item_cover_image_overlay_icon_wrap
per);

        let queue_overlay_content_item_cover_image_overlay_icon = document.createElement("i");
        queue_overlay_content_item_cover_image_overlay_icon.classList.add("fa-solid");
        queue_overlay_content_item_cover_image_overlay_icon.classList.add("fa-play");
        queue_overlay_content_item_cover_image_overlay_icon.classList.add("queue-content-item-cover-overlay-
icon");

queue_overlay_content_item_cover_image_overlay_icon_wrapper.appendChild(queue_overlay_content_item_co
ver_image_overlay_icon);

        // queue overlay item name and artists
        let queue_overlay_content_item_name_and_artists_column = document.createElement("div");
        queue_overlay_content_item_name_and_artists_column.classList.add("queue-content-item-name-and-
artists-column");
        queue_overlay_content_item.appendChild(queue_overlay_content_item_name_and_artists_column);

        let queue_overlay_content_item_name = document.createElement("p");
        queue_overlay_content_item_name.classList.add("queue-content-item-name-row");
```

```
        queue_overlay_content_item_name.innerText = song_name;
        queue_overlay_content_item_name_and_artists_column.appendChild(queue_overlay_content_item_name);

        let queue_overlay_content_item_artists = document.createElement("p");
        queue_overlay_content_item_artists.classList.add("queue-content-item-artists-row");
        queue_overlay_content_item_artists.innerText = song_artist_names;
        queue_overlay_content_item_name_and_artists_column.appendChild(queue_overlay_content_item_artists);

        // queue overlay item duration
        let queue_overlay_content_item_duration = document.createElement("p");
        queue_overlay_content_item_duration.classList.add("queue-content-item-duration-column");
        queue_overlay_content_item_duration.innerText = return_duration_string(song_duration);
        queue_overlay_content_item.appendChild(queue_overlay_content_item_duration);

        // adding clone of queue overlay item to player extended overlay queue part
        let player_extended_queue_content_item = queue_overlay_content_item.cloneNode(true);
        player_extended_queue_content_item.id = "player-extended-queue-content-item-song-" + song_id;
        player_extended_queue_content_wrapper.appendChild(player_extended_queue_content_item);

        if (song_id == current_song_id){
            queue_overlay_content_item.scrollIntoView();
            player_extended_queue_content_item.scrollIntoView();
        }

        // add event listeners to play that song
        queue_overlay_content_item.onclick = () => play_song(song_id);
        player_extended_queue_content_item.onclick = () => play_song(song_id);
    }
}

function render_lyrics(song_id){
    song_id = Number(song_id);
    floating_lyrics_overlay_content_wrapper.innerHTML = songs_obj[song_id].song_lyrics_markup;
    player_extended_lyrics_content_wrapper.innerHTML = songs_obj[song_id].song_lyrics_markup;
    floating_lyrics_overlay_content_wrapper.scrollTop = 0;
    player_extended_lyrics_content_wrapper.scrollTop = 0;
}

function profile_dropdown_open_close(open = true){
    if (profile_dropdown_overlay.classList.contains("invisible-element") && open){
        profile_dropdown_overlay.classList.remove("shrunk-element");
        profile_dropdown_overlay.classList.remove("invisible-element");
    }
    else {
        profile_dropdown_overlay.classList.add("shrunk-element");
        profile_dropdown_overlay.classList.add("invisible-element");
    }
```

```
}

function login_logout(){
    if (current_user_id) logout();
    else login_popup_open_close();
}

function login_popup_open_close(open = true){
    login_popup.classList.toggle("invisible-element");
    switch_login_popup_tab();
}

function switch_login_popup_tab(tab = "login"){
    if (tab == "login"){
        login_popup_login_tab_button.classList.add("login-popup-tab-active-button");
        login_popup_signup_tab_button.classList.remove("login-popup-tab-active-button");
        login_popup_login_tab.classList.remove("invisible-element");
        login_popup_login_tab.classList.remove("shrunk-element");
        login_popup_signup_tab.classList.add("invisible-element");
        login_popup_signup_tab.classList.add("shrunk-element");
        setTimeout(() => {
            login_popup_login_tab.querySelector(".login-popup-username-input").focus();
        }, 200);
    } else {
        login_popup_login_tab_button.classList.remove("login-popup-tab-active-button");
        login_popup_signup_tab_button.classList.add("login-popup-tab-active-button");
        login_popup_signup_tab.classList.remove("invisible-element");
        login_popup_signup_tab.classList.remove("shrunk-element");
        login_popup_login_tab.classList.add("invisible-element");
        login_popup_login_tab.classList.add("shrunk-element");
        setTimeout(() => {
            login_popup_signup_tab.querySelector(".login-popup-username-input").focus();
        }, 200);
    }
    login_popup_login_tab.querySelector(".login-popup-username-caution-text").classList.add("display-none");
    login_popup_login_tab.querySelector(".login-popup-password-caution-text").classList.add("display-none");
    login_popup_signup_tab.querySelector(".login-popup-username-caution-text").classList.add("display-none");
    login_popup_signup_tab.querySelector(".login-popup-password-caution-text").classList.add("display-none");
    login_popup_signup_tab.querySelector(".login-popup-email-caution-text").classList.add("display-none");
    login_popup_signup_tab.querySelector(".login-popup-confirm-password-caution-text").classList.add("display-
none");
}

function login(){
    let username_input = login_popup_login_tab.querySelector(".login-popup-username-input");
    let username_caution_text = login_popup_login_tab.querySelector(".login-popup-username-caution-text");
    let password_input = login_popup_login_tab.querySelector(".login-popup-password-input");
```

```
    let password_caution_text = login_popup_login_tab.querySelector(".login-popup-password-caution-text");

    let username = username_input.value;
    let password = password_input.value;

    // checking if username and password are entered
    if (! username || ! password){
        username_input.focus();
        username_caution_text.classList.add("display-none");
        password_caution_text.classList.remove("display-none");
        password_caution_text.innerText = "Please enter all the details";
        return;
    } else {
        username_caution_text.classList.add("display-none");
        password_caution_text.classList.add("display-none");
    }

    // checking if username exists
    let user_id = Object.keys(users_obj).filter(id => users_obj[id].username == username)[0];
    let user_obj = users_obj[user_id];
    if (! user_obj){
        username_input.focus();
        username_caution_text.classList.remove("display-none");
        username_caution_text.innerText = "Username does not exist. Please try again or sign up";
        return;
    }
    // checking if password is correct
    if (user_obj.password != password){
        password_input.focus();
        password_caution_text.classList.remove("display-none");
        password_caution_text.innerText = "Incorrect password. Please try again";
        return;
    }

    // if everything is correct then login i.e. set current_user_id and update user content
    current_user_id = user_id;

    username_input.value = "";
    password_input.value = "";

    update_user_content(true);
    login_popup_open_close(false);
    make_floating_notification("login");
}

function signup(){
    let username_input = login_popup_signup_tab.querySelector(".login-popup-username-input");
```

```
    let username_caution_text = login_popup_signup_tab.querySelector(".login-popup-username-caution-text");
    let email_input = login_popup_signup_tab.querySelector(".login-popup-email-input");
    let email_caution_text = login_popup_signup_tab.querySelector(".login-popup-email-caution-text");
    let password_input = login_popup_signup_tab.querySelector(".login-popup-password-input");
    let password_caution_text = login_popup_signup_tab.querySelector(".login-popup-password-caution-text");
    let confirm_password_input = login_popup_signup_tab.querySelector(".login-popup-confirm-password-input");
    let confirm_password_caution_text = login_popup_signup_tab.querySelector(".login-popup-confirm-password-caution-text");

    let username = username_input.value;
    let email = email_input.value;
    let password = password_input.value;
    let confirm_password = confirm_password_input.value;

    if (! username || ! email || ! password || ! confirm_password){
        username_input.focus();
        confirm_password_caution_text.classList.remove("display-none");
        confirm_password_caution_text.innerText = "Please enter all the details";
        return;
    } else {
        username_caution_text.classList.add("display-none");
        email_caution_text.classList.add("display-none");
        password_caution_text.classList.add("display-none");
    }
    if (Object.keys(users_obj).filter(id => users_obj[id].username == username).length > 0){
        username_input.focus();
        username_caution_text.classList.remove("display-none");
        username_caution_text.innerText = "Username taken. Please try again";
        return;
    } else {
        email_caution_text.classList.add("display-none");
        password_caution_text.classList.add("display-none");
        confirm_password_caution_text.classList.add("display-none");
    }
    if (email_regex.test(email) == false){
        email_input.focus();
        email_caution_text.classList.remove("display-none");
        email_caution_text.innerText = "Please enter a valid email";
        return;
    } else {
        email_caution_text.classList.add("display-none");
        password_caution_text.classList.add("display-none");
        confirm_password_caution_text.classList.add("display-none");
    }
    if (password != confirm_password){
        password_input.focus();
```

```
        confirm_password_caution_text.classList.remove("display-none");
        confirm_password_caution_text.innerText = "Passwords do not match. Please try again";
        return;
    } else {
        confirm_password_caution_text.classList.add("display-none");
    }

    let new_user = {
        "username": username,
        "email": email,
        "password": password,
        "created_at": new Date(),
        "liked_song_ids_list": [],
        "saved_playlist_ids_list": [],
        "recently_played_song_ids_list": [],
        "recently_played_playlist_ids_list": []
    }

    username_input.value = "";
    email_input.value = "";
    password_input.value = "";
    confirm_password_input.value = "";

    current_user_id = Object.keys(users_obj).length + 1;
    sessionStorage.setItem("current_user_id", current_user_id);
    users_obj[current_user_id] = new_user;
    localStorage.setItem("users_obj", JSON.stringify(users_obj));

    update_user_content(true);
    login_popup_open_close(false);
    make_floating_notification("signup");
}

function logout(){
    current_user_id = null;

    update_user_content();
    make_floating_notification("logout");
}

function update_user_obj(){
    let user_obj = users_obj[current_user_id];

    user_obj.liked_song_ids_list = liked_song_ids_list;
    user_obj.saved_playlist_ids_list = saved_playlist_ids_list;
    user_obj.recently_played_song_ids_list = recently_played_song_ids_list;
    user_obj.recently_played_playlist_ids_list = recently_played_playlist_ids_list;
```

```
    users_obj[current_user_id] = user_obj;

    localStorage.setItem("users_obj", JSON.stringify(users_obj));
}

function update_user_content(just_logged_in = false){

    users_obj = localStorage.getItem("users_obj") ? JSON.parse(localStorage.getItem("users_obj")) : {};

    if (current_user_id) {

        let user_obj = users_obj[current_user_id];

        if (just_logged_in){
            profile_dropdown_overlay_login_status.innerText = "Logged into @" +
users_obj[current_user_id].username;
            profile_dropdown_overlay_login_logout_button.innerText = "Logout";

            sessionStorage.setItem("current_user_id", JSON.stringify(current_user_id));

            liked_song_ids_list = user_obj.liked_song_ids_list;
            saved_playlist_ids_list = user_obj.saved_playlist_ids_list;
            recently_played_song_ids_list = user_obj.recently_played_song_ids_list;
            recently_played_playlist_ids_list = user_obj.recently_played_playlist_ids_list;
        }

        update_user_obj();

    } else {

        profile_dropdown_overlay_login_status.innerText = "Not logged in";
        profile_dropdown_overlay_login_logout_button.innerText = "Login / Sign Up";

        liked_song_ids_list = [];
        saved_playlist_ids_list = [];
        recently_played_song_ids_list = [];
        recently_played_playlist_ids_list = [];
        sessionStorage.setItem("current_user_id", JSON.stringify(current_user_id));
    }
    if (current_content_window_content_type == "tab"){
        render_content_window(current_content_window_content_tag_id, true);
    }
    // else render_playlist(current_content_window_content_tag_id);
}

function like_unlike_song(song_id){
```

```
    if (!current_user_id){
        make_floating_notification("not_logged_in", false);
        return;
    }

    song_id = Number(song_id);

    let playlist_liked_unliked_item_like_icon_id = "#playlist-content-item-like-button-icon-song-" + song_id;
    let playlist_current_playing_item_like_icon_element =
content_window.querySelector(playlist_liked_unliked_item_like_icon_id);

    // Adding or removing song id from liked_song_ids_list
    if (liked_song_ids_list.includes(song_id)){
        liked_song_ids_list = liked_song_ids_list.filter(id => id != song_id);
        make_floating_notification("unlike_song");
        if (song_id == current_song_id){
            player_like_button_icon.classList.remove("fa-solid");
            player_like_button_icon.classList.add("fa-regular");
        }
        if (playlist_current_playing_item_like_icon_element){
            playlist_current_playing_item_like_icon_element.classList.remove("fa-solid");
            playlist_current_playing_item_like_icon_element.classList.add("fa-regular");
        }
    } else {
        liked_song_ids_list.unshift(song_id);
        make_floating_notification("like_song");
        if (song_id == current_song_id){
            player_like_button_icon.classList.remove("fa-regular");
            player_like_button_icon.classList.add("fa-solid");
        }
        if (playlist_current_playing_item_like_icon_element){
            playlist_current_playing_item_like_icon_element.classList.add("fa-solid");
            playlist_current_playing_item_like_icon_element.classList.remove("fa-regular");
        }
    }
    update_user_content();
}

function shuffle_on_off(){
    shuffle = !shuffle;
    sessionStorage.setItem("shuffle", shuffle);
    if (shuffle){
        player_shuffle_button_icon.classList.remove("inactive-button");
    } else {
        player_shuffle_button_icon.classList.add("inactive-button");
    }
}
```

```
function play_previous_song(){
    // if current time is greater than 5 seconds, then play the current song from the beginning
    if (player_audio_controls.currentTime > 5){
        player_audio_controls.currentTime = 0;
        return;
    }
    if (shuffle){
        let queue_previous_song_index = queue_played_song_indexes_list.pop();
        if (! queue_previous_song_index) return;
        let previous_song_id = queue_song_id_list[queue_previous_song_index];
        play_song(previous_song_id);
    } else {
        // if shuffle off, current time is less than 5 seconds, and queue not at beginning then play the previous song
        if (queue_current_song_index > 0){
            queue_current_song_index--;
            let previous_song_id = queue_song_id_list[queue_current_song_index];
            play_song(previous_song_id);
        }
    }
    // if shuffle off, current time is less than 5 seconds, and queue at beginning then do nothing
}

function toggle_play_pause_icon(play){
    if (play){
        player_play_pause_button_icon.classList.remove("fa-circle-play");
        player_play_pause_button_icon.classList.add("fa-circle-pause");
    }
    else{
        player_play_pause_button_icon.classList.remove("fa-circle-pause");
        player_play_pause_button_icon.classList.add("fa-circle-play");
    }
}

function play_pause(play = false){
    if (player_play_pause_button_icon.classList.contains("fa-circle-play") || play){
        player_audio_controls.play();
        player_play_pause_button_container.title = "Pause";
    }
    else{
        player_audio_controls.pause();
        player_play_pause_button_container.title = "Play";
    }
}

function play_song(song_id, playlist_id = null, append_queue = false, shuffle_current_queue = false){
    song_id = Number(song_id);
```

```
    if (first_play){
        nav.classList.remove("full-height");
        right_window.classList.remove("full-height");
        player.classList.remove("inactive-player");
    }
    recently_played_song_ids_list = recently_played_song_ids_list.filter(id => id != song_id);
    recently_played_song_ids_list.unshift(song_id);
    if (playlist_id){
        playlist_id = Number(playlist_id);
        recently_played_playlist_ids_list = recently_played_playlist_ids_list.filter(id => id != song_id);
        recently_played_playlist_ids_list.unshift(song_id);
    }

    // if the song is already playing, do nothing
    if (song_id == current_song_id){
        return;
    }

    // updating queue current song id for current queue index to use when rendering new songs from tabs
    let last_song_id = current_song_id;
    current_song_id = song_id;

    let song_obj = songs_obj[song_id];

    // update player song info
    let img_src = "./media/covers/" + songs_obj[song_id].song_file_name_root + ".jpg";
    player_song_cover_image.src = img_src;
    player_extended_cover_image.src = img_src;
    player_extended_cover_image.alt = "song_cover_image";
    player_song_name.innerText = song_obj.song_name;
    player_song_artist.innerText = song_obj.song_artist_names;
    player_audio_total_duration_label.innerText = return_duration_string(song_obj.song_duration);

    render_lyrics(song_id);

    if (liked_song_ids_list.includes(song_id)){
        player_like_button_icon.classList.add("fa-solid");
        player_like_button_icon.classList.remove("fa-regular");
    }
    else{
        player_like_button_icon.classList.remove("fa-solid");
        player_like_button_icon.classList.add("fa-regular");
    }

    player_audio_controls.innerHTML = `<source src="./media/audios/${song_obj.song_file_name_root}.mp3"
type="audio/mpeg">`;
    player_audio_controls.load();
```

```
    play_pause(true);

    if (playlist_id != null){
        update_queue(playlist_id, false, append_queue, shuffle_current_queue);
    }
    else{
        update_queue(song_id, true);
    }
    highlight_playing_item(last_song_id, current_song_id);
    first_play = false;

    update_user_content();
}

function play_next_song(){
    // if repeat == 2 i.e. repeat one, play the same song again
    if (repeat == 2){
        player_audio_controls.currentTime = 0;
        play_pause(true);
        return;
    }
    // if repeat == 1 or 0 i.e. repeat all or no repeat
    queue_played_song_indexes_list.push(queue_current_song_index);
    queue_current_song_index++;
    // if repeat == 1 i.e. repeat all, move to first song in queue if last song is playing
    if (repeat == 1){
        if (shuffle){
            // repeat == 1 and shuffle on so moving to random old song in queue if last song is playing
            if (queue_played_song_indexes_list.length == queue_song_id_list.length){
                queue_played_song_indexes_list = [];
            }
            queue_current_song_index = Math.floor(Math.random() * queue_song_id_list.length);
            while (queue_played_song_indexes_list.includes(queue_current_song_index)){
                queue_current_song_index = Math.floor(Math.random() * queue_song_id_list.length);
            }
        }
        else{
            // repeat == 1 and shuffle off so moving to first song in queue if last song is playing
            if (queue_current_song_index == queue_song_id_list.length){
                queue_current_song_index = 0;
            }
        }
        let next_song_id = queue_song_id_list[queue_current_song_index];
        play_song(next_song_id);
        return;
    }
    // if repeat == 0 i.e. no repeat
```

```
    if (shuffle){
        // repeat == 0 and shuffle on so appending next random playlist to queue if last song is playing
        if (queue_played_song_indexes_list.length == queue_song_id_list.length){
            let current_playlist_id = songs_obj[current_song_id].song_playlist_ids[0];
            let next_random_playlist_id = Math.floor(Math.random() * Object.keys(playlists_obj).length);
            while (next_random_playlist_id == current_playlist_id){
                next_random_playlist_id = Math.floor(Math.random() * Object.keys(playlists_obj).length);
            }
            let next_random_playlist_song_ids = playlists_obj[next_random_playlist_id].playlist_song_ids;
            let next_random_song_index = Math.floor(Math.random() * next_random_playlist_song_ids.length);
            let next_song_id = next_random_playlist_song_ids[next_random_song_index];
            play_song(next_song_id, next_random_playlist_id, true);
            return;
        }
        queue_current_song_index = Math.floor(Math.random() * queue_song_id_list.length);
        while (queue_played_song_indexes_list.includes(queue_current_song_index)){
            queue_current_song_index = Math.floor(Math.random() * queue_song_id_list.length);
        }
        let next_song_id = queue_song_id_list[queue_current_song_index];
        play_song(next_song_id);
        return;
    }
    // if repeat == 0 and no shuffle, append next playlist in queue if last song is playing
    if (queue_current_song_index == queue_song_id_list.length){
        let current_playlist_id = songs_obj[current_song_id].song_playlist_ids[0];
        let next_playlist_id = (current_playlist_id + 1) % Object.keys(playlists_obj).length;
        let next_song_id = playlists_obj[next_playlist_id].playlist_song_ids[0];
        play_song(next_song_id, next_playlist_id, true);
        return;
    }
    let next_song_id = queue_song_id_list[queue_current_song_index];
    play_song(next_song_id);
}

function repeat_toggle(){
    repeat = ++repeat % 3;
    sessionStorage.setItem("repeat", repeat);
    if (repeat == 0){
        player_repeat_button_icon.classList.add("inactive-button");
        player_repeat_button_icon.classList.remove("small-button");
        player_repeat_button_icon_one.classList.add("shrunk-element");
    }
    else if (repeat == 1){
        player_repeat_button_icon.classList.remove("inactive-button");
    }
    else{
        player_repeat_button_icon.classList.remove("inactive-button");
```

```
        player_repeat_button_icon.classList.add("small-button");
        player_repeat_button_icon_one.classList.remove("shrunk-element");
    }
}

function options_dialog_open_close(){
    // open or close options dialog
}

function queue_overlay_open_close(open = true){
    if (player_queue_button_icon.classList.contains("inactive-button") && open){
        player_queue_button_icon.classList.remove("inactive-button");
        floating_queue_overlay.classList.remove("shrunk-element");
        player_floating_part.style.height = player_floating_part_full_height;
    }
    else{
        player_queue_button_icon.classList.add("inactive-button");
        floating_queue_overlay.classList.add("shrunk-element");
        if (floating_lyrics_overlay.classList.contains("shrunk-element")){
            player_floating_part.style.height = 0;
        }
    }
}

function lyrics_overlay_open_close(open = true){
    if (player_lyrics_button_icon.classList.contains("fa-regular") && open){
        floating_lyrics_overlay.classList.remove("shrunk-element");
        player_floating_part.style.height = player_floating_part_full_height;
        player_lyrics_button_icon.classList.remove("inactive-button");
        player_lyrics_button_icon.classList.add("fa-solid");
        player_lyrics_button_icon.classList.remove("fa-regular");
    }
    else{
        floating_lyrics_overlay.classList.add("shrunk-element");
        if (floating_queue_overlay.classList.contains("shrunk-element")){
            player_floating_part.style.height = 0;
        }
        player_lyrics_button_icon.classList.add("inactive-button");
        player_lyrics_button_icon.classList.remove("fa-solid");
        player_lyrics_button_icon.classList.add("fa-regular");
    }
}

function update_volume_icon(){
    player_volume_button_icon.classList = "fa-solid";
    if (player_audio_controls.volume == 0 || player_audio_controls.muted){
        player_volume_button_icon.classList.add("fa-volume-xmark");
```

```
    }
    else{
        player_volume_button_icon.classList.add("fa-volume-high");
    }
}

function mute_unmute(){
    if (player_audio_controls.muted){
        player_audio_controls.muted = false;
        player_volume_button_container.title = "Mute";
    }
    else{
        player_audio_controls.muted = true;
        player_volume_button_container.title = "Unmute";
    }
    update_volume_icon();
}

function update_volume(){
    player_audio_controls.volume = player_volume_seek_bar.value / 100;
    update_volume_icon();
}

function highlight_playing_item(last_song_id, next_song_id){
    last_song_id = Number(last_song_id);
    next_song_id = Number(next_song_id);
    let playlist_last_playing_item = "#playlist-content-item-song-" + last_song_id;
    let queue_overlay_last_playing_item = "#queue-overlay-content-item-song-" + last_song_id;
    let player_extended_queue_last_playing_item = "#player-extended-queue-content-item-song-" + last_song_id;

    let playlist_last_playing_item_element = content_window.querySelector(playlist_last_playing_item);
    let queue_overlay_last_playing_item_element = floating_queue_overlay_content_wrapper.querySelector(queue_overlay_last_playing_item);
    let player_extended_queue_last_playing_item_element = player_extended_queue_content_wrapper.querySelector(player_extended_queue_last_playing_item);

    let last_playing_item_elements = [playlist_last_playing_item_element, queue_overlay_last_playing_item_element, player_extended_queue_last_playing_item_element];
    last_playing_item_elements.forEach((element) => {
        if (element){
            element.classList.remove("active-item");
            let playing_item_overlay_icon_wrapper = element.querySelector(".content-item-cover-overlay-icon-wrapper");
            let playing_item_overlay_icon = element.getElementsByTagName("i")[0];
            playing_item_overlay_icon_wrapper.classList.remove("opacity-1");
            playing_item_overlay_icon.classList.remove("fa-volume-high");
            playing_item_overlay_icon.classList.add("fa-play");
```

```
        }
    });

    let playlist_current_playing_item = "#playlist-content-item-song-" + next_song_id;
    let queue_overlay_current_playing_item = "#queue-overlay-content-item-song-" + next_song_id;
    let player_extended_queue_current_playing_item = "#player-extended-queue-content-item-song-" + next_song_id;

    let playlist_current_playing_item_element = content_window.querySelector(playlist_current_playing_item);
    let queue_overlay_current_playing_item_element = floating_queue_overlay_content_wrapper.querySelector(queue_overlay_current_playing_item);
    let player_extended_queue_current_playing_item_element = player_extended_queue_content_wrapper.querySelector(player_extended_queue_current_playing_item);

    let current_playing_item_elements = [playlist_current_playing_item_element, queue_overlay_current_playing_item_element, player_extended_queue_current_playing_item_element];
    current_playing_item_elements.forEach((element) => {
        if (element){
            element.classList.add("active-item");
            let playing_item_overlay_icon_wrapper = element.querySelector(".content-item-cover-overlay-icon-wrapper");
            let playing_item_overlay_icon = element.getElementsByTagName("i")[0];
            playing_item_overlay_icon_wrapper.classList.add("opacity-1");
            playing_item_overlay_icon.classList.add("fa-volume-high");
            playing_item_overlay_icon.classList.remove("fa-play");
        }
    });
}

function toggle_extended_player(extend = true){
    if (player_extended_overlay.classList.contains("shrunk-element") && extend){
        player_extended_overlay.classList.remove("shrunk-element");
        content_window.classList.add("invisible-element");
        queue_overlay_open_close(false);
        lyrics_overlay_open_close(false);
        player_queue_button_container.classList.add("shrunk-element");
        player_queue_button_container.classList.add("invisible-element");
        player_lyrics_button_container.classList.add("shrunk-element");
        player_lyrics_button_container.classList.add("invisible-element");
        player_extend_button_icon.classList.add("rotate-180");
        shrink_unshrink_nav_bar(true);
    } else {
        player_extended_overlay.classList.add("shrunk-element");
        content_window.classList.remove("invisible-element");
        player_queue_button_container.classList.remove("shrunk-element");
        player_queue_button_container.classList.remove("invisible-element");
        player_lyrics_button_container.classList.remove("shrunk-element");
```

```
        player_lyrics_button_container.classList.remove("invisible-element");
        player_extend_button_icon.classList.remove("rotate-180");
        shrink_unshrink_nav_bar(false);
    }
}

function shrink_unshrink_nav_bar(shrink = true){
    if (!nav_shrinking_items[0].classList.contains("shrunk-element") && shrink){
        nav_shrinking_items.forEach((nav_shrinking_item) => {
            nav_shrinking_item.classList.add("shrunk-element");
            nav_shrinking_item.classList.add("invisible-element");
        });
        nav_button_containers.forEach(nav_button_container => nav_button_container.classList.add("nav-button-
container-shrunk"));
        root.style.setProperty('--nav-width', '4.5rem');
        root.style.setProperty('--nav-min-width', 'unset');
        root.style.setProperty('--nav-max-width', 'unset');
    }
    else {
        nav_shrinking_items.forEach(nav_shrinking_item => {
            nav_shrinking_item.classList.remove("shrunk-element");
            nav_shrinking_item.classList.remove("invisible-element");
        });
        nav_button_containers.forEach(nav_button_container => nav_button_container.classList.remove("nav-
button-container-shrunk"));
        root.style.setProperty('--nav-width', nav_original_width);
        root.style.setProperty('--nav-min-width', nav_original_min_width);
        root.style.setProperty('--nav-max-width', nav_original_max_width);
    }
}

function make_floating_notification(type, show_icon = true){

    let message = "";
    if (type == "welcome"){
        message = "Welcome to Re□Ound!";
    }

    else if (type == "not_logged_in"){
        message = "You must be logged in to do that!";
    }

    else if (type == "login"){
        message = "Logged into @" + users_obj[current_user_id].username;
    } else if (type == "logout"){
        message = "Logged out!";
    } else if (type == "signup"){
```

```
    message = "Signed up with @" + users_obj[current_user_id].username + "!";
}

else if (type == "like_song"){
    message = "Added to Liked Songs";
} else if (type == "unlike_song"){
    message = "Removed from Liked Songs";
}

else if (type == "add_playlist_to_library"){
    message = "Added Playlist to Library";
} else if (type == "remove_playlist_from_library"){
    message = "Removed Playlist from Library";
} else if (type == "add_playlist_to_queue"){
    message = "Added Playlist to Queue";
}

else if (type == "shuffle_on"){
    message = "Shuffle On";
} else if (type == "shuffle_off"){
    message = "Shuffle Off";
} else if (type == "repeat_none"){
    message = "Repeat Off";
} else if (type == "repeat_all"){
    message = "Repeat All";
} else if (type == "repeat_one"){
    message = "Repeat One";
}

else if (type == "add_song_to_queue"){
    message = "Added Song to Queue";
}

let floating_notification = document.createElement("div");
floating_notification.classList.add("floating-notification");
body.appendChild(floating_notification);

if (show_icon){
    let floating_notification_icon = document.createElement("i");
    floating_notification_icon.classList.add("fa-solid");
    floating_notification_icon.classList.add("fa-square-check");
    floating_notification_icon.classList.add("floating-notification-icon");
    floating_notification.appendChild(floating_notification_icon);
}

let floating_notification_text = document.createElement("p");
floating_notification_text.classList.add("floating-notification-text");
```

```
    floating_notification_text.innerHTML = message;
    floating_notification.appendChild(floating_notification_text);

    setTimeout(() => {
        floating_notification.classList.add("invisible-element");
        setTimeout(() => {
            floating_notification.remove();
        }, 500);
    }, 3000);
}


// Event Listeners
nav.onmouseenter = () => shrink_unshrink_nav_bar(false);
nav.onmouseleave = () => {
    if (! player_extended_overlay.classList.contains("shrunk-element"))
        shrink_unshrink_nav_bar(true);
}

nav_home_button_container.onclick = () => render_content_window("home");

nav_explore_button_container.onclick = () => render_content_window("explore");

nav_library_button_container.onclick = () => render_content_window("library");

search_bar.onfocus = show_hide_search_suggestions;
search_bar.onkeyup = update_search_suggestions;
search_bar.onkeydown = (event) => {
    if (event.key === "Enter"){
        event.preventDefault();
        search_suggestions_wrapper.classList.add("shrunk-element");
        search_bar.blur();
        render_content_window("search");
    }
}

profile_button_container.onclick = (event) => {
    event.stopPropagation();
    profile_dropdown_open_close();
}

profile_dropdown_overlay_login_logout_button.onclick = login_logout;

login_popup_login_tab_button.onclick = () => switch_login_popup_tab();
login_popup_signup_tab_button.onclick = () => switch_login_popup_tab("signup");
login_popup_close_button.onclick = login_popup_open_close;
login_popup_submit_login_button.onclick = login;
```

```
login_popup_submit_signup_button.onclick = signup;

player_like_button_container.onclick = () => like_unlike_song(current_song_id);

player_shuffle_button_container.onclick = shuffle_on_off;

player_previous_song_button_container.onclick = play_previous_song;

player_play_pause_button_container.onclick = () => play_pause();
player_extended_cover_image.onclick = () => play_pause();

player_next_song_button_container.onclick = play_next_song;

player_repeat_button_container.onclick = repeat_toggle;

player_options_button_container.onclick = options_dialog_open_close;

player_queue_button_container.onclick = queue_overlay_open_close;

player_lyrics_button_container.onclick = lyrics_overlay_open_close;

player_volume_button_container.onclick = mute_unmute;

player_volume_seek_bar.oninput = () => update_volume();

player_extend_button_container.onclick = toggle_extended_player;

document.onkeydown = (event) => {
    if (! login_popup.classList.contains("invisible-element")) {
        if (event.key == "Escape"){
            event.preventDefault();
            login_popup_open_close(false);
        }
        if (event.key == "Enter"){
            event.preventDefault();
            if (login_popup_login_tab_button.classList.contains("login-popup-tab-active-button")){
                login();
            }
            else signup();
        }
        return;
    };
    if (event.key == "Escape"){
        if (document.activeElement.tagName == "INPUT" && document.activeElement.type == "text"){
            event.preventDefault();
            event.target.blur();
        }
```

```
        else {
            event.preventDefault();
            toggle_extended_player();
        }
    }
    if (document.activeElement.tagName == "INPUT" && (document.activeElement.type == "text" || document.activeElement.type == "password" || document.activeElement.type == "email")){
        return;
    }
    if (event.key == " "){
        event.preventDefault();
        play_pause();
    }
    if (event.key == "ArrowLeft"){
        event.preventDefault();
        player_audio_controls.currentTime -= 5;
    }
    if (event.key == "ArrowRight"){
        event.preventDefault();
        player_audio_controls.currentTime += 5;
    }
    if (event.key == "ArrowUp"){
        event.preventDefault();
        player_volume_seek_bar.value = Number(player_volume_seek_bar.value) + 5;
        update_volume();
    }
    if (event.key == "ArrowDown"){
        event.preventDefault();
        player_volume_seek_bar.value = Number(player_volume_seek_bar.value) - 5;
        update_volume();
    }
    if (event.key.toUpperCase() == "S"){
        event.preventDefault();
        shuffle_on_off();
    }
    if (event.key.toUpperCase() == "R"){
        event.preventDefault();
        repeat_toggle();
    }
    if (event.key.toUpperCase() == "M"){
        event.preventDefault();
        mute_unmute();
    }
    if (event.key.toUpperCase() == "Q" && player_extended_overlay.classList.contains("shrunk-element")){
        event.preventDefault();
        queue_overlay_open_close();
    }
```

```
    if (event.key.toUpperCase() == "L" && player_extended_overlay.classList.contains("shrunk-element")){
        event.preventDefault();
        lyrics_overlay_open_close();
    }
    if (event.shiftKey && event.key.toUpperCase() == "P"){
        event.preventDefault();
        play_previous_song();
    }
    if (event.shiftKey && event.key.toUpperCase() == "N"){
        event.preventDefault();
        play_next_song();
    }
    if (event.ctrlKey && event.key.toUpperCase() == "F"){
        event.preventDefault();
        search_bar.focus();
    }
}
document.onclick = (event) => {
    if (document.activeElement.id != "search-bar"){
        show_hide_search_suggestions(false);
    }
    if (event.target != profile_dropdown_overlay_login_status && event.target != profile_dropdown_overlay){
        profile_dropdown_open_close(false);
    }
}

player_audio_controls.onplay = () => toggle_play_pause_icon(true);
player_audio_controls.onpause = () => toggle_play_pause_icon(false);

player_audio_controls.onloadedmetadata = () => player_audio_controls.currentTime = 0;

player_audio_controls.onended = play_next_song;

player_audio_controls.ontimeupdate = update_audio_seek_bar_and_time_stamp;

player_audio_seek_bar.oninput = () => player_audio_controls.currentTime = player_audio_seek_bar.value;


// Function Calls
render_content_window("home");
render_nav_library_content();

update_user_content(true);
```