

1.0 Introduction

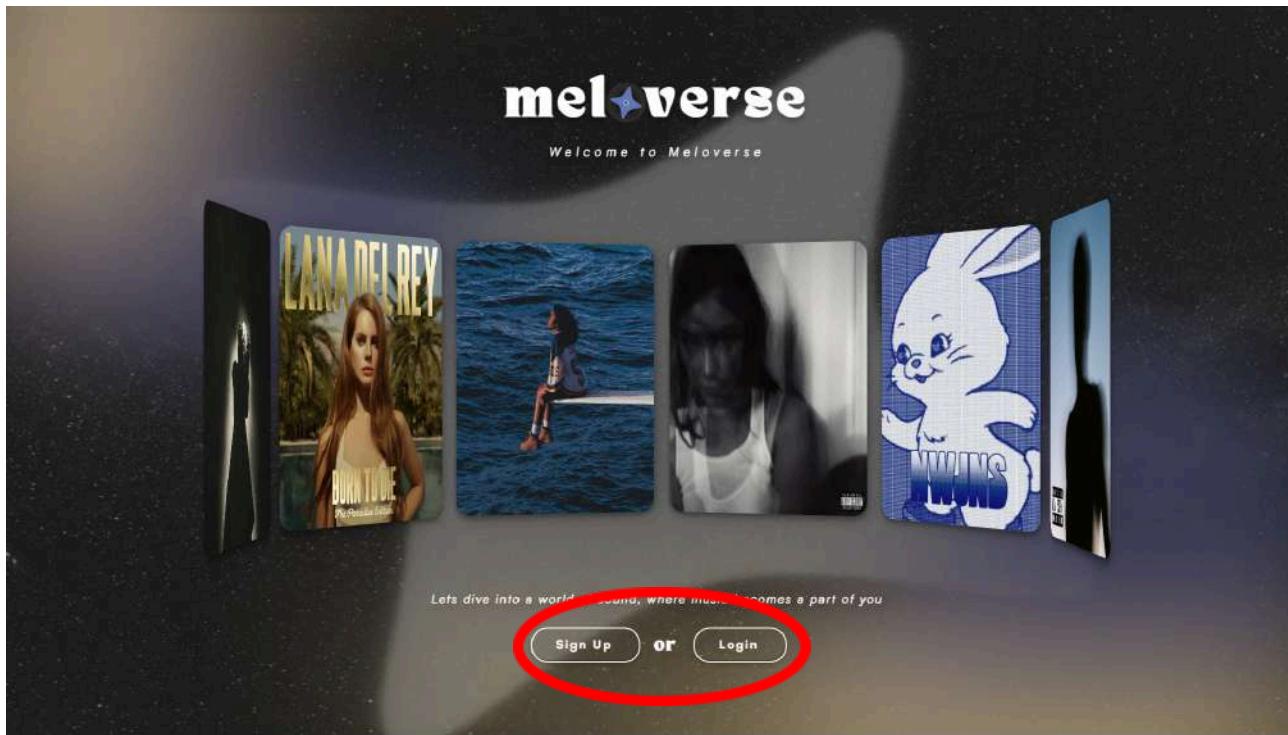
In a world where music is more than just sound, it is a way to connect, express, and inspire, Meloverse emerges as a transformative music streaming application. Designed with a user-first approach, Meloverse goes beyond traditional music platforms by blending personalization, engagement, and an immersive connection between listeners and their favorite artists. With its thoughtfully crafted features, Meloverse ensures that music isn't just heard—it's experienced.

One of the standout features of Meloverse is the Points Page, a rewards-driven system where users can earn points through their interactions on the platform and redeem them for amazing rewards. This gamified approach makes engaging with music even more exciting and rewarding. Additionally, the Artist Page provides a beautifully designed space where users can explore an artist's world. From their popular songs and albums to exclusive merchandise, this page offers a comprehensive introduction to the musicians users love.

These features, combined with a personalized experience that tailors recommendations based on individual preferences, set Meloverse apart from other platforms. By offering real-time updates on concerts and new releases, behind-the-scenes content, and exclusive artist interactions, Meloverse strengthens the bond between users and their favorite musicians. Whether you're discovering new music or deepening your connection with familiar artists, every interaction on Meloverse is crafted to feel meaningful and engaging.

With its innovative features and intuitive design, Meloverse isn't just a music streaming platform—it's a dynamic space where music becomes a lifestyle. It's a platform that rewards passion, celebrates artistry, and redefines what it means to connect with music in the digital age. Welcome to Meloverse, where every song, every artist, and every moment matters.

2.0 User Manual



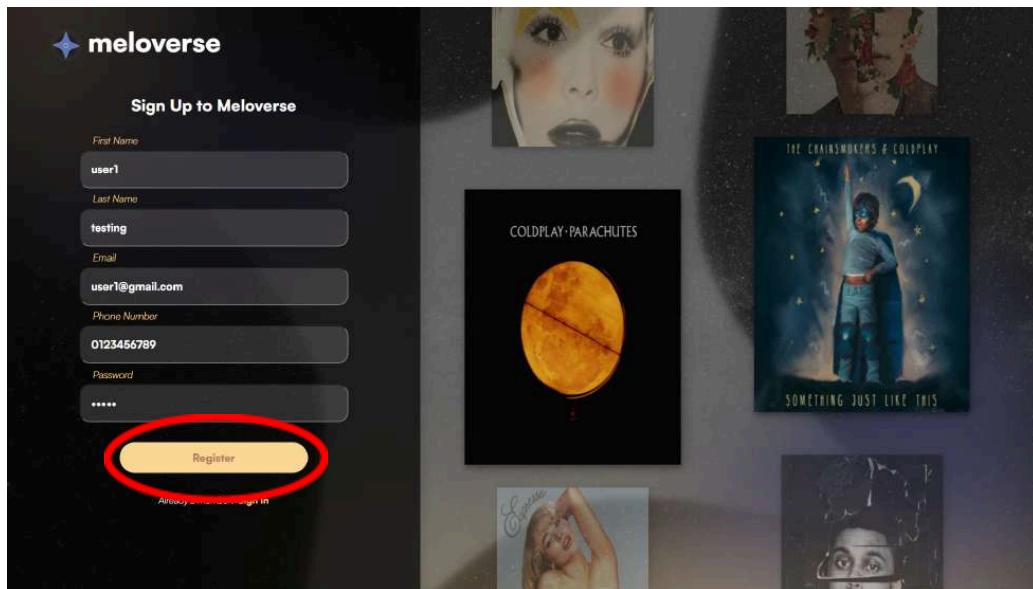
When you first visit the website, you will land on our Landing Page with two buttons: **Sign Up** and **Login**.

If You Don't Have an Account: Click the **Sign Up** button to create a new account.

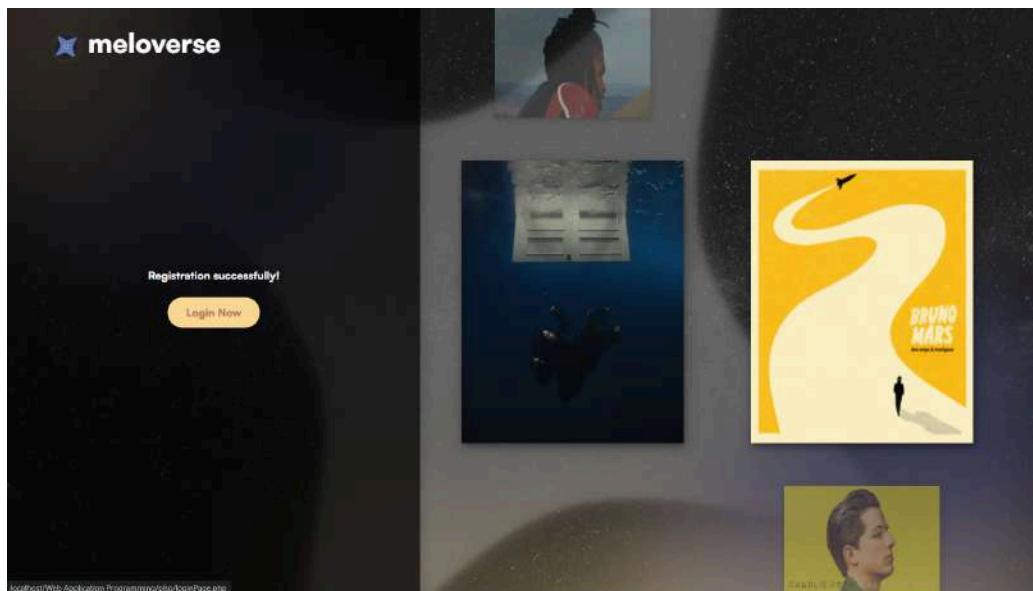
If You Already Have an Account: Click the **Login** button to access your account.

For New Users:

If you don't have an account, click the Sign Up button. You will then be redirected to the Sign Up page. Then, you have to fill in the required information to create an account and access our website. Once you have filled in the information, click the Register button.

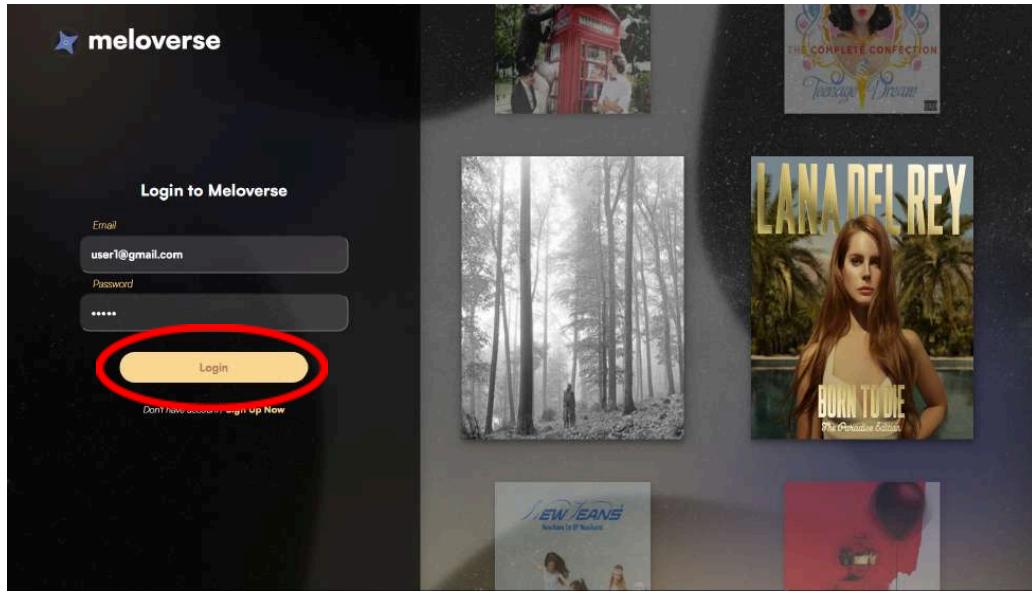


After clicking the register button, a message (Registration successfully!) will pop up, and you are ready to go.

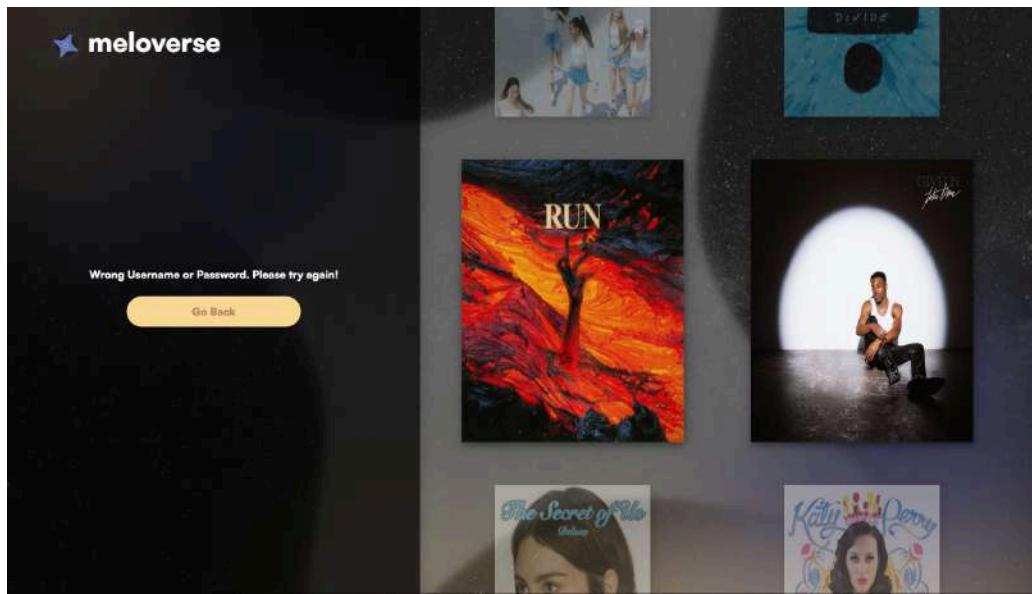


For Existing Users:

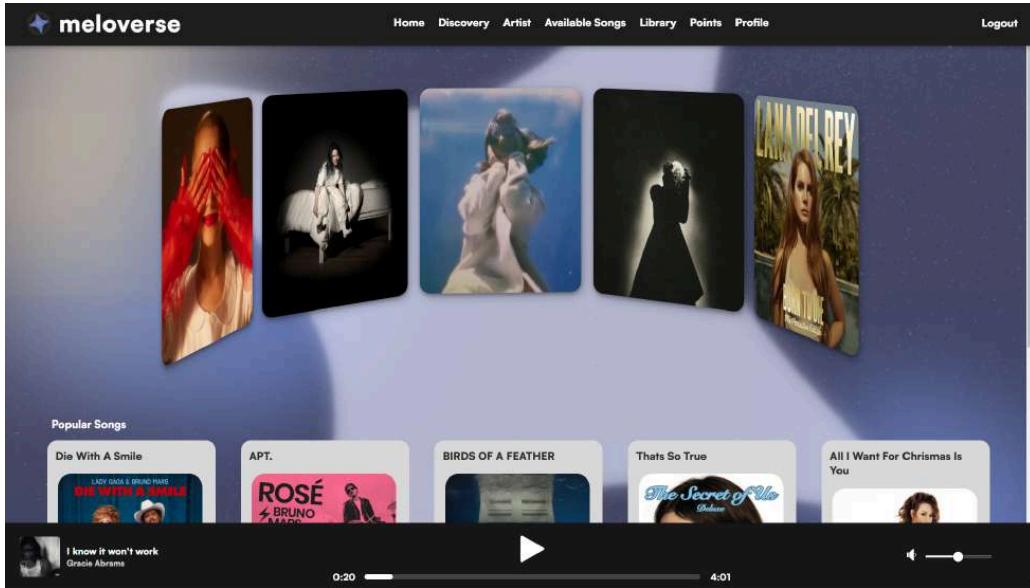
If you click 'Login,' you will be redirected to our login page. On this page, you will need to fill in your account credentials and, once done, click '**Login**' and you will be successfully logged in. However, if you do not already have an account, click '**Sign Up Now**' to be redirected to the Sign Up page.



If you enter the wrong information, you will see an error message: '**Wrong Username or Password. Please try again!**' and will need to re-enter your credentials.

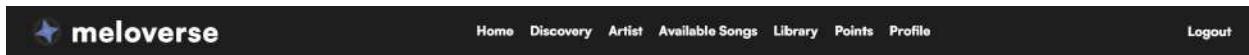


After logging in, you will arrive at the Meloverse home page. On the home page, you can view two sections: our selection of **popular songs** and **popular artists**. At the top, you will see a header section, and at the bottom, a footer section. Our header and footer will be present at every page.



Header Section:

From the header section, you can access different pages such as **Home**, **Discovery**, **Artists**, **Library**, **Available Songs**, **Points**, and **Profile** by clicking on the titles.

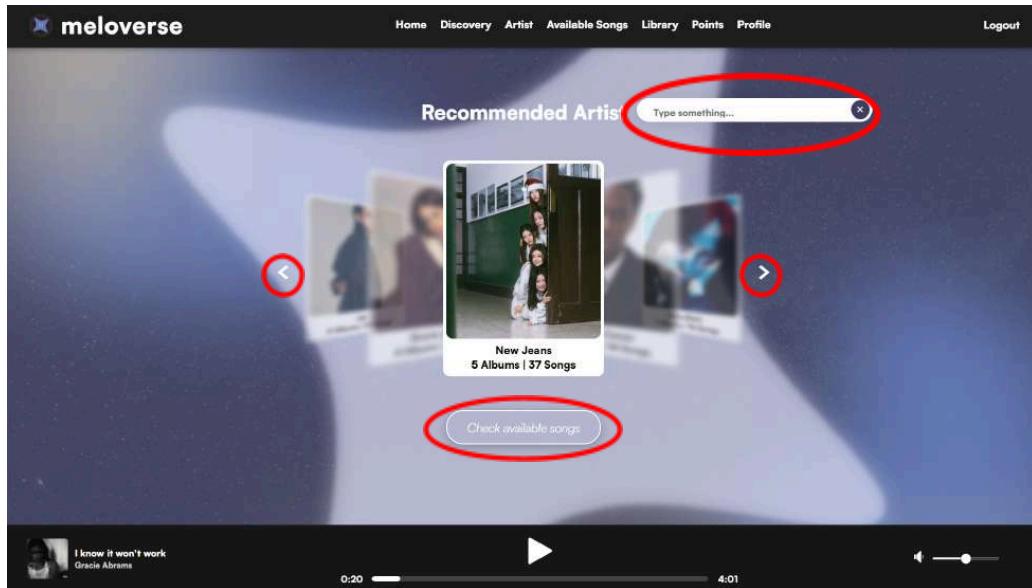


Footer Section:

From the footer section, you can access the song controller, which includes features such as the pause/play button, volume slider, and progress bar (allowing you to click and adjust the song's duration).

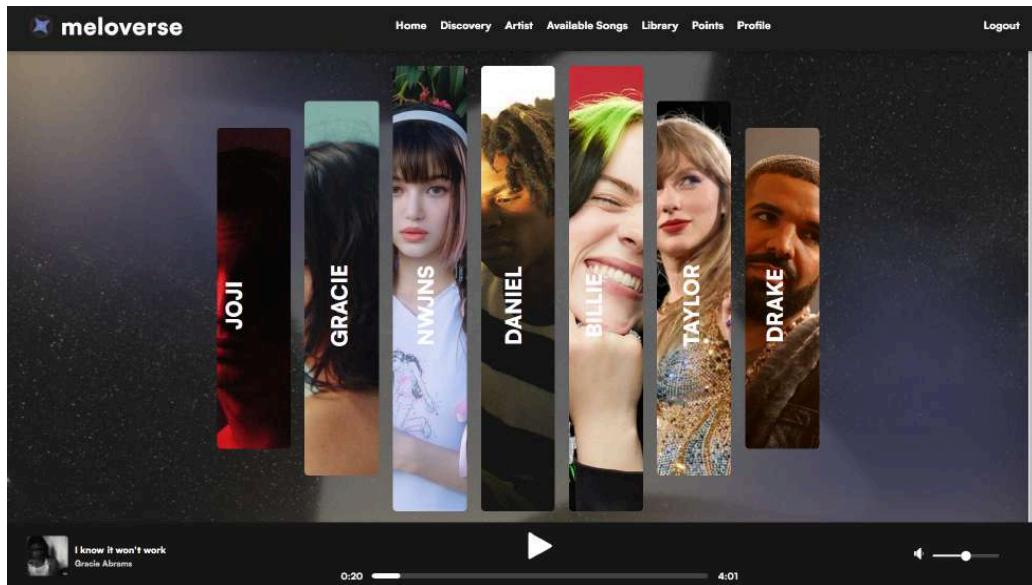


By clicking '**Discovery**' in the header, you will be redirected to the Discovery Page. There, you can browse our recommended artists and search for your favorite artist using the **search bar**. To scroll through our Recommended Artists, we have **arrow buttons** on either side.

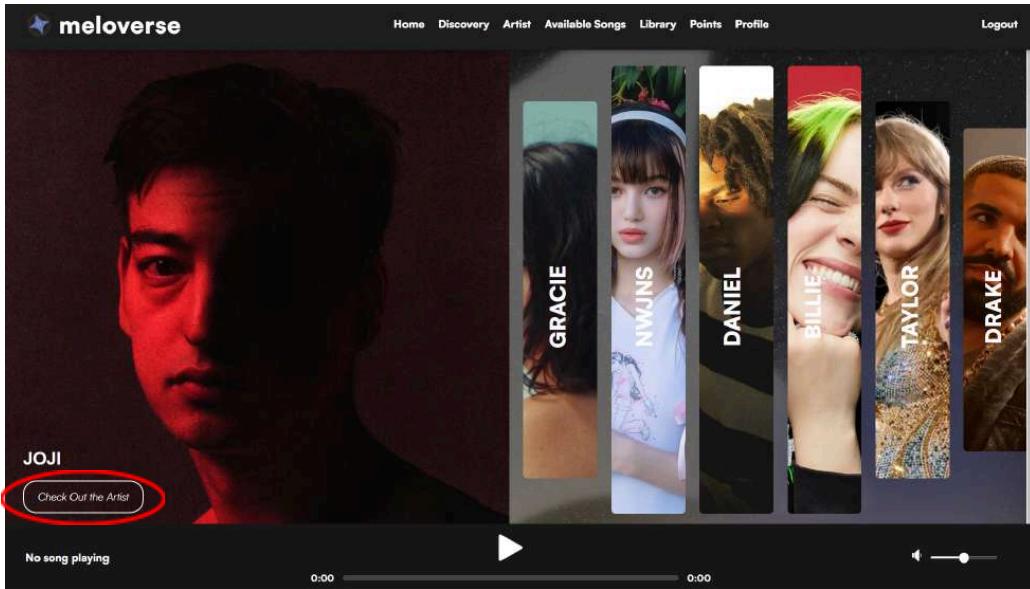


To check out the available songs for the current artist, click the '**Check Available Songs**' button. You will be directed to our Available Songs List page.

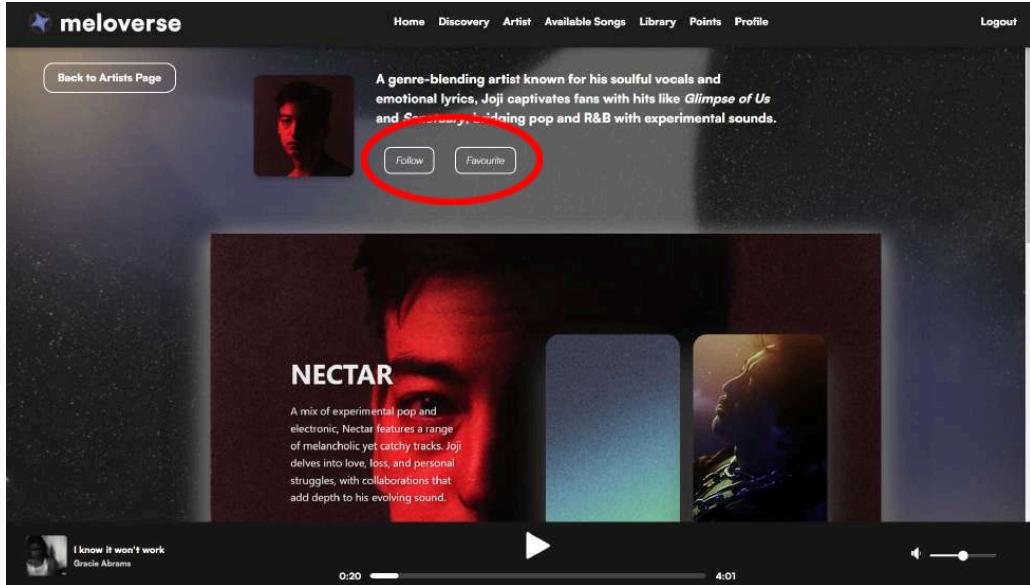
By clicking '**Artist**' in the header, you will be redirected to the Artist Page. Here you can view information of your favourite artist by clicking on their icons.



For example, by clicking Joji.



To view more information of ‘**Joji**’, click ‘**Check Out the Artist**’ button to be redirected to the artist information page, here you can ‘**Follow**’ or ‘**Favourite**’ the artists, check out album information, view soundtracks and merchandise.



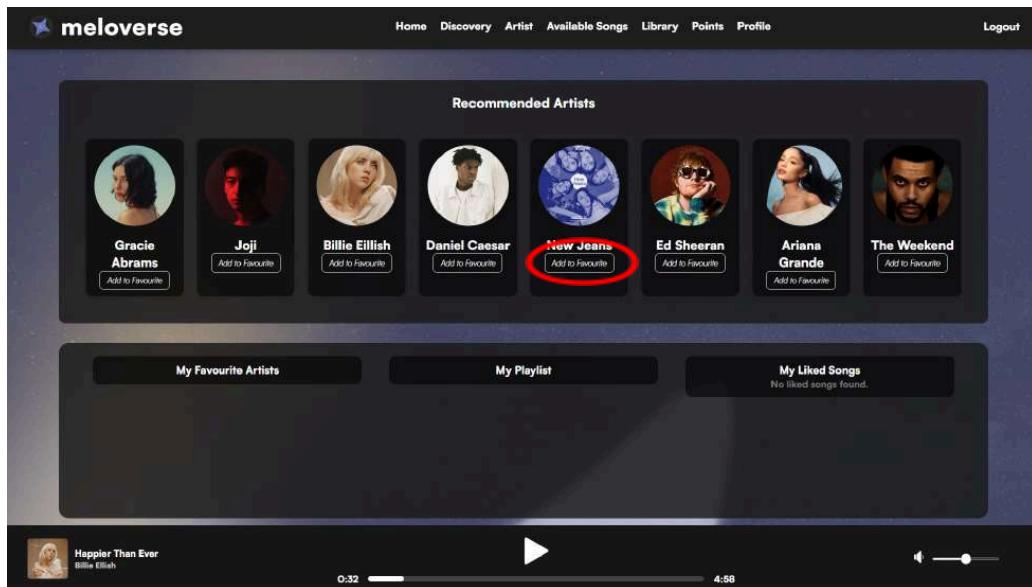
Clicking ‘Available Songs’ in the header redirects you to the **Available Song List** page, where you can browse songs. Each song has three buttons: **Add to Queue**, **Add to Liked Songs**, and **Add to Playlist** to manage your music easily.

The screenshot shows the 'Available Songs' section for the artist Billie Eilish. It lists three songs: 'Happier Than Ever', 'HIT ME HARD AND SOFT', and 'WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?'. Each song entry includes a small thumbnail, the song title, and three circular buttons for managing the queue. The first song, 'Happier Than Ever', has its management buttons circled in red at the bottom. Below the list is a player bar showing the currently playing song, 'I know it won't work' by Gracie Abrams, with a play/pause button and volume controls.

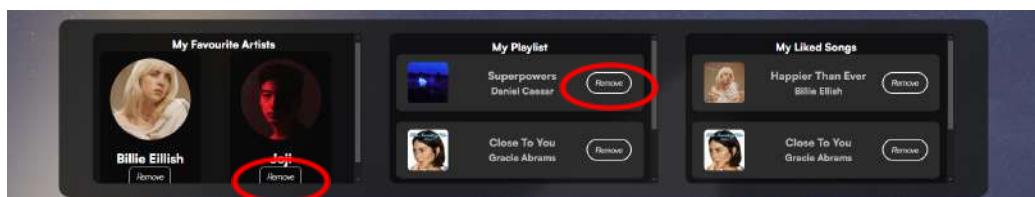
After clicking on a song title, you will be directed to the **Song Page**, where the selected music will start playing, and the **queue list** will be displayed. You can manage your queue by using the ‘Remove’ button to remove songs from the list. The currently playing song will be displayed in the footer at the bottom of the page, where you can also control your music playback, volume and play/pause.

The screenshot shows the 'Song Page' for the song 'I Love You, I'm Sorry' by Gracie Abrams. On the left, there's a large image of a vinyl record with a portrait of the artist. The song title and artist name are displayed above the record. In the center, there's a waveform visualization. On the right, a sidebar titled 'Queue' lists the current queue of songs, each with a 'Remove' button. The first song in the queue is 'WILDFLOWER' by Billie Eilish, with its remove button circled in red. At the bottom of the page, there's a player bar showing the currently playing song, 'I Love You, I'm Sorry' by Gracie Abrams, with a play/pause button and volume controls.

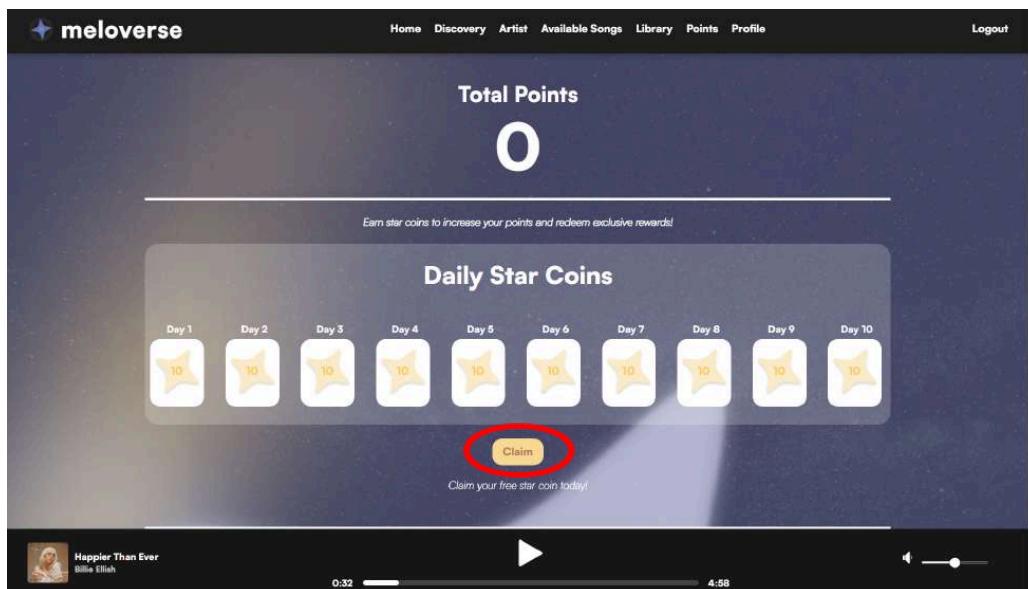
To access the Library Page, click on the ‘**Library**’ option in the header, which will redirect you to the page which displays **your favourite artists, liked songs, and playlists**. Below each artist profile, there is an ‘**Add Favourite**’ button; by clicking this button, the artist’s profile will be added to the ‘My Favourite Artist’ section.



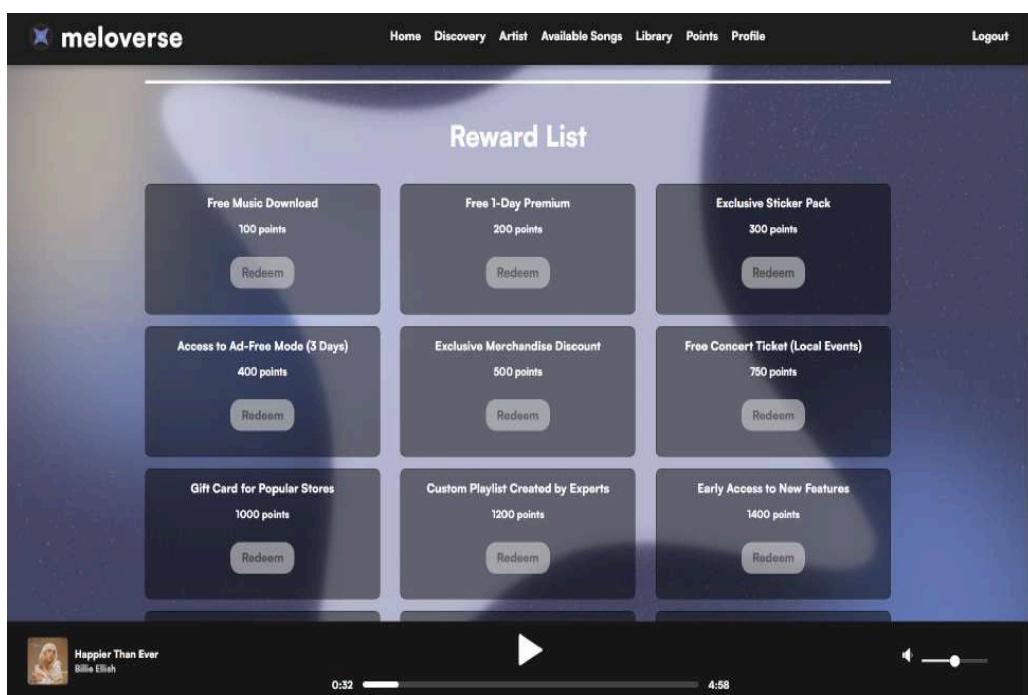
As you add songs to your **playlist** and **liked songs**, a ‘**Remove**’ button will appear, allowing you to remove these songs if desired. Similarly, favourite artists can also be removed using the ‘**Remove**’ button.



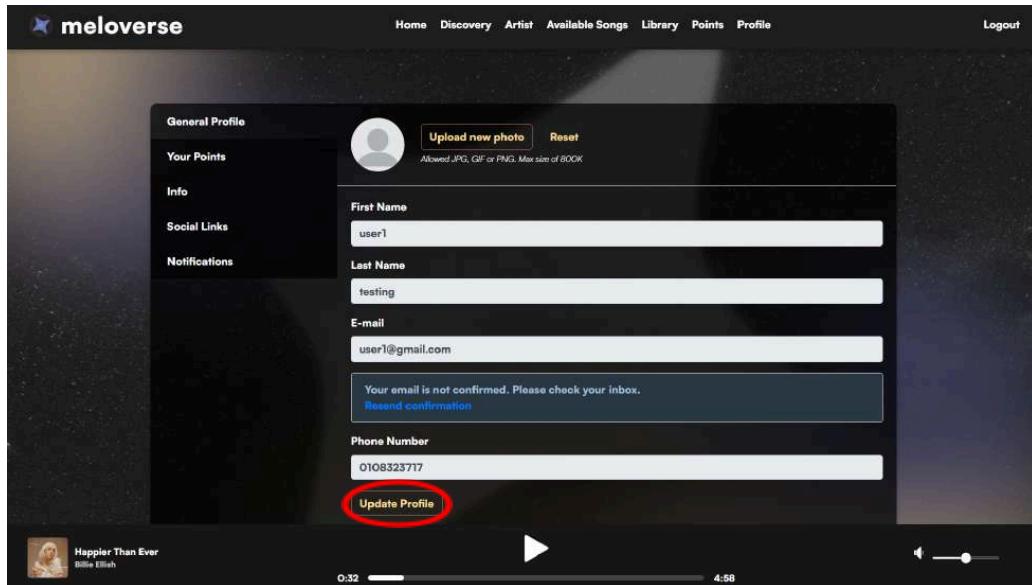
By clicking '**Points**' in the header, you will be redirected to the Points Page. This is where your points will be displayed, along with the points system and rewards system. By clicking the '**Claim**' button, you will be awarded 10 points. You can only claim more points by logging in everyday. At the same time, the daily star coins for Day 1 will be shown as claimed.



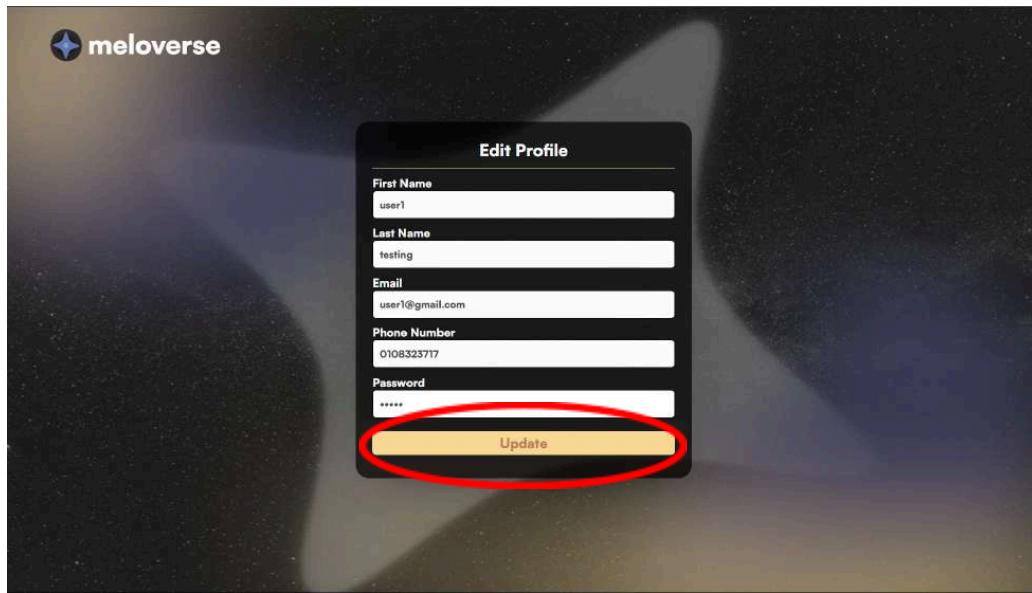
By accumulating points day by day, you can redeem amazing rewards from the rewards list below.



By clicking '**Profile**' in the header, you will be redirected to the Profile Page. This is where your user information will be displayed. To update your information, click the '**Update Profile**' button.

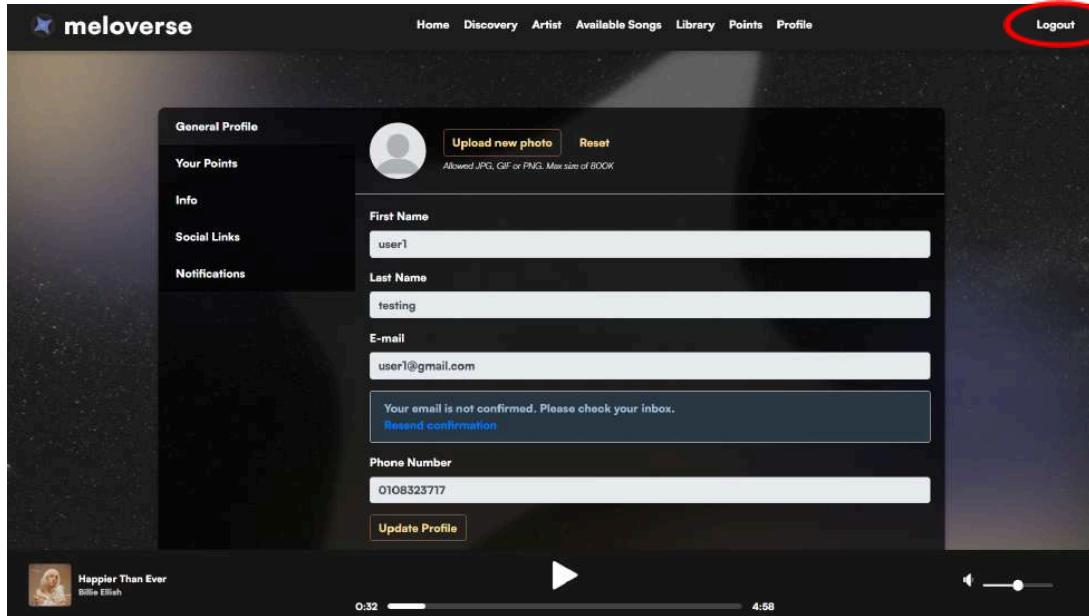


By clicking the '**Update Profile**' button, you will be redirected to the Update Profile Page.

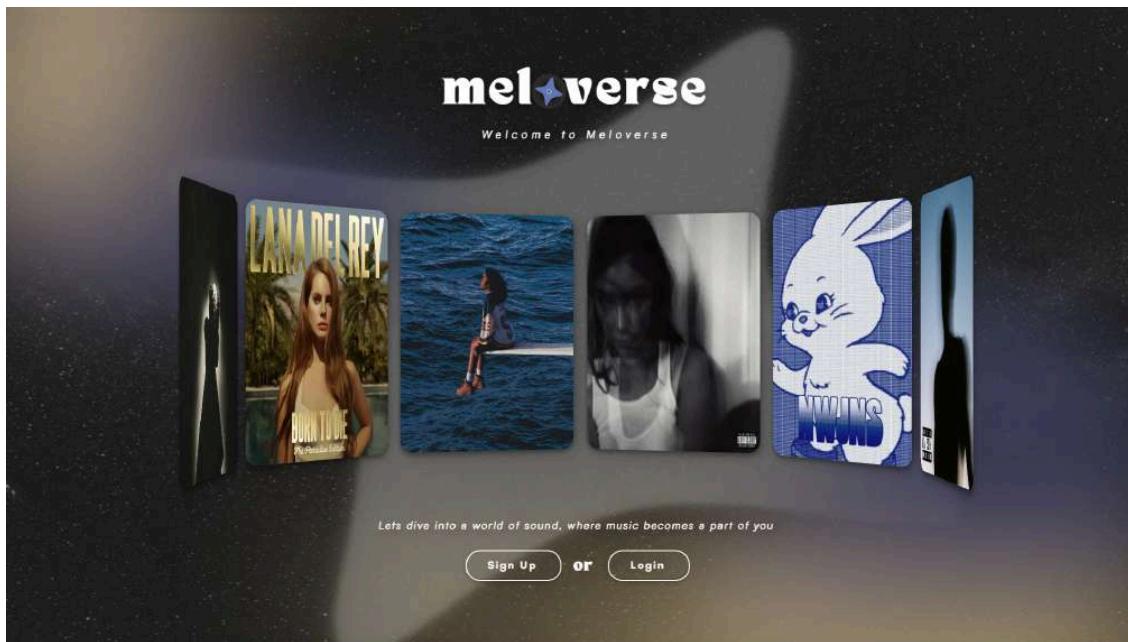


Once you have finished updating your profile information, click the '**Update**' button. You will then be redirected back to the Profile Page.

Lastly, when you're done enjoying or listening to music on the website, you can click '**Logout**' from the header section at the top right corner of the page in order to logout.



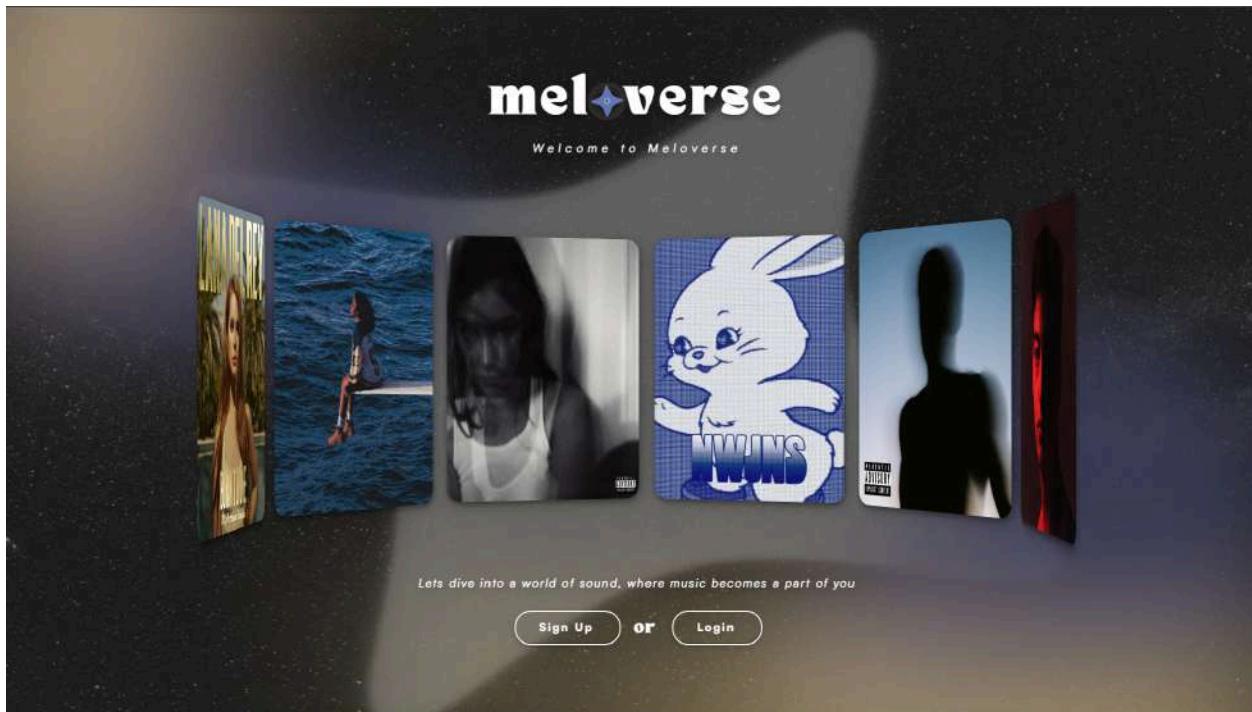
After clicking the Logout button, you will be redirected to the Landing Page, where you will have to re-login if you want to access the website again.



3.0 Pages and Designs

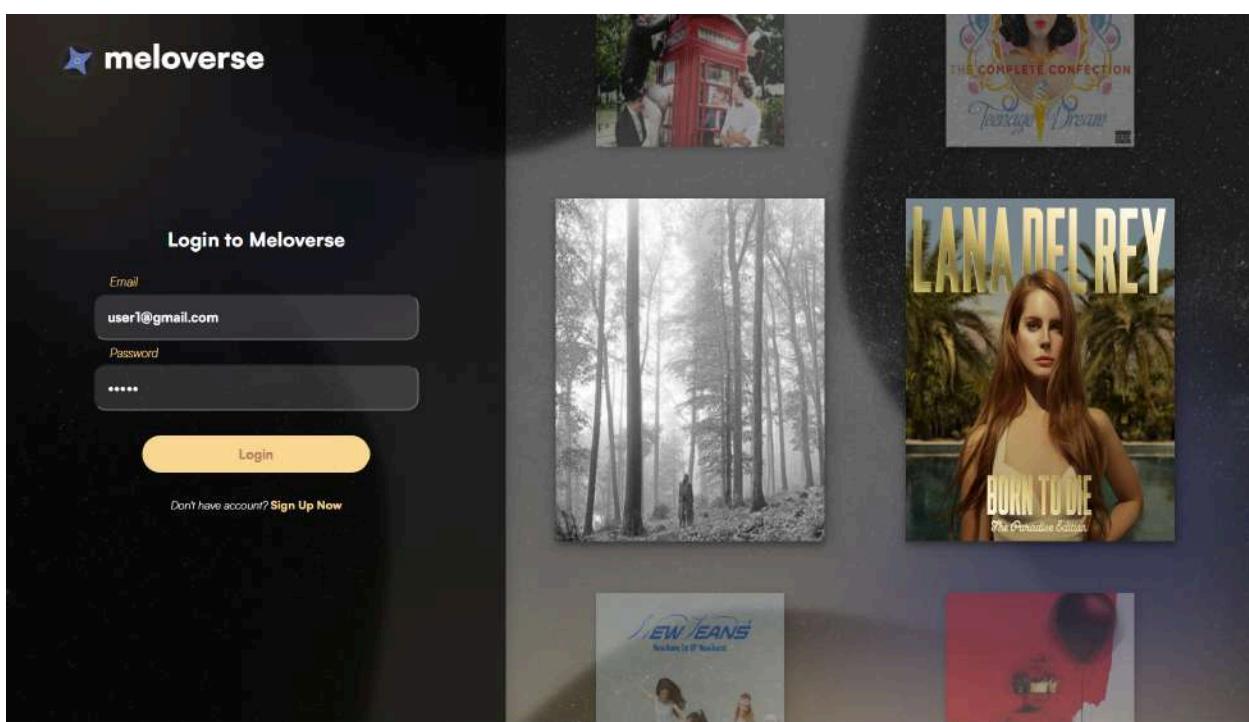
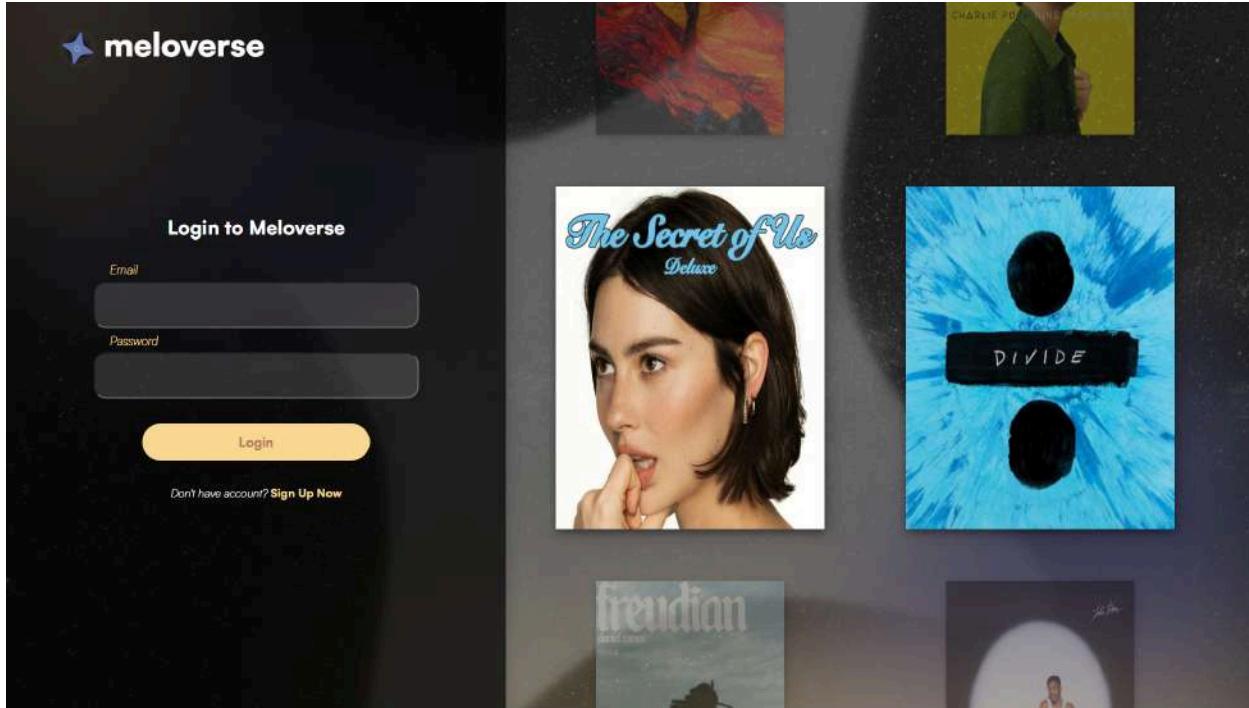
3.1 Landing Page

The Landing Page serves as the first impression of Meloverse, welcoming users with an engaging design and a brief overview of the platform. It provides easy navigation to sign up or log in, ensuring a smooth user entry.



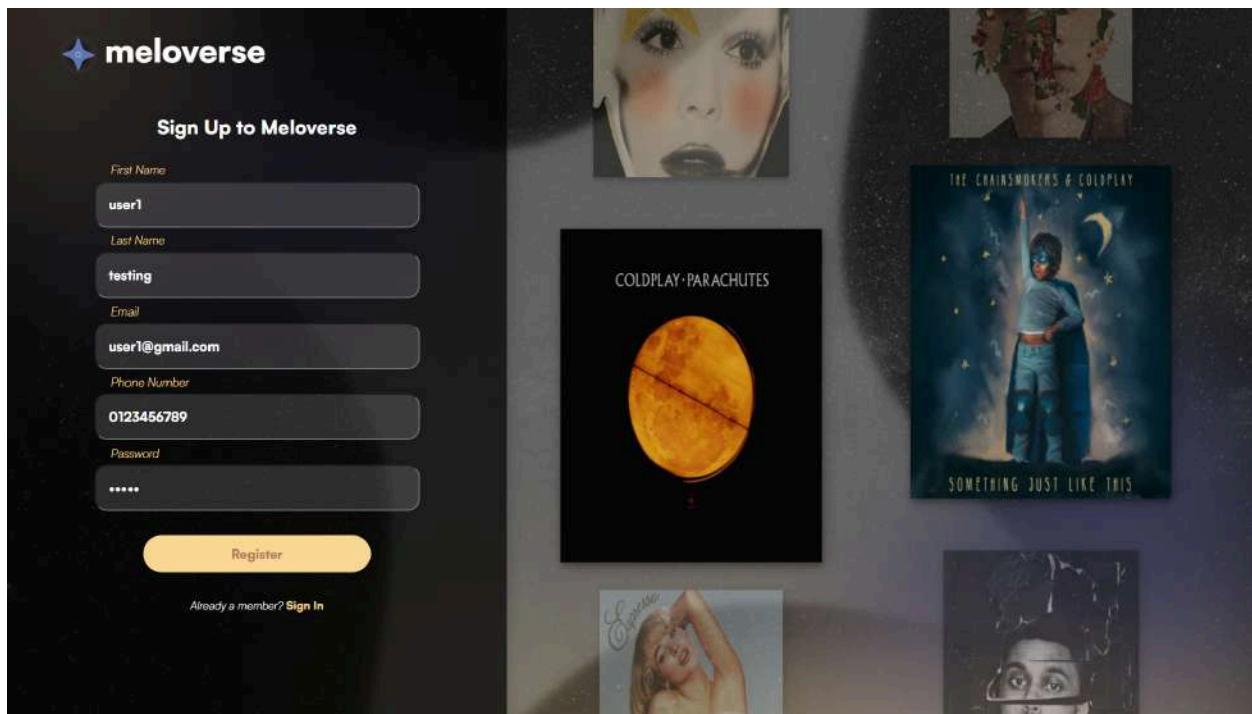
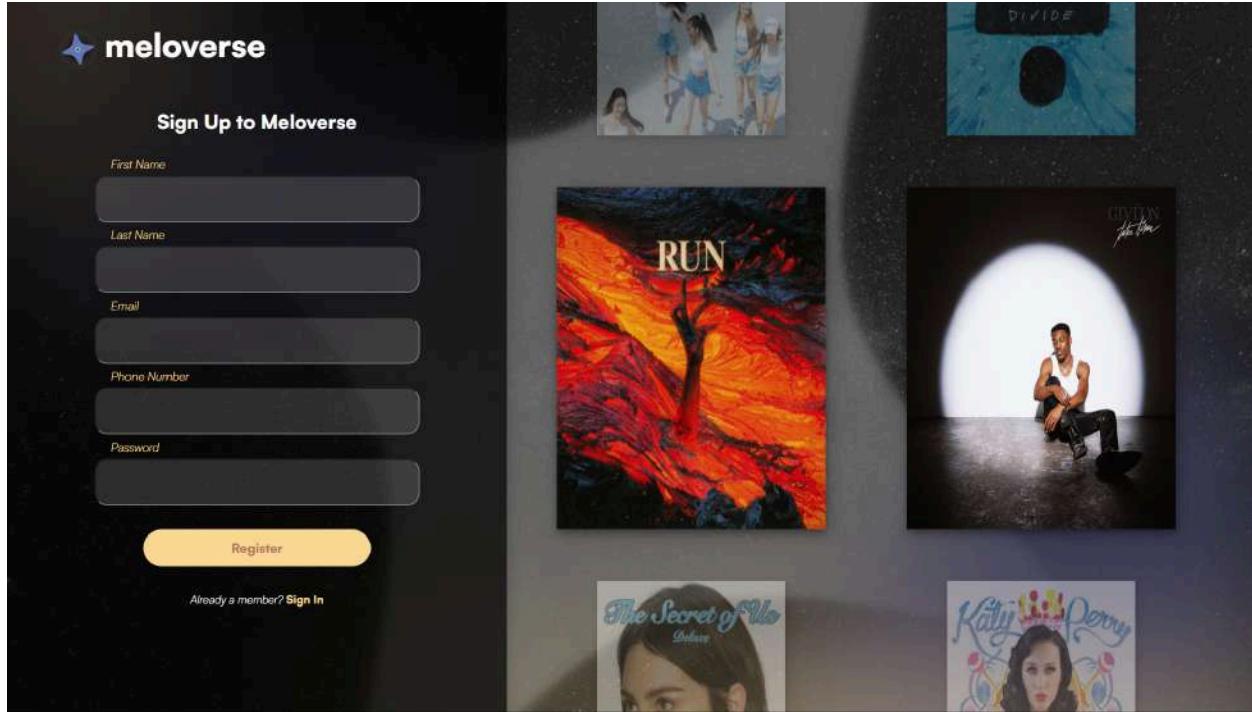
3.2 Login Page

The Login Page allows registered users to securely access their accounts using their credentials. It ensures data security while offering a simple and clean layout for quick access.



3.3 Sign Up Page

The Sign-Up Page enables new users to register on Meloverse. With a user-friendly form, it collects essential details to create personalized user accounts.



3.4 Home Page

The Home Page is the central hub for users, featuring personalized recommendations, trending songs, and highlights. It provides quick access to artists, albums, and content tailored to user preferences.

meloverse

Home Discovery Artist Available Songs Library Points Profile Logout

Popular Songs

- Die With A Smile
- ROSE
- BIRDS OF A FEATHER
- Thats So True
- All I Want For Chrismas Is You

I know it won't work
Gracie Abrams

0:20 4:01

meloverse

Home Discovery Artist Available Songs Library Points Profile Logout

Popular Songs

- Die With A Smile
- ROSE
- BIRDS OF A FEATHER
- Thats So True
- All I Want For Chrismas Is You

Taylor Swift

Bruno Mars

Lady Gaga

The Weekend

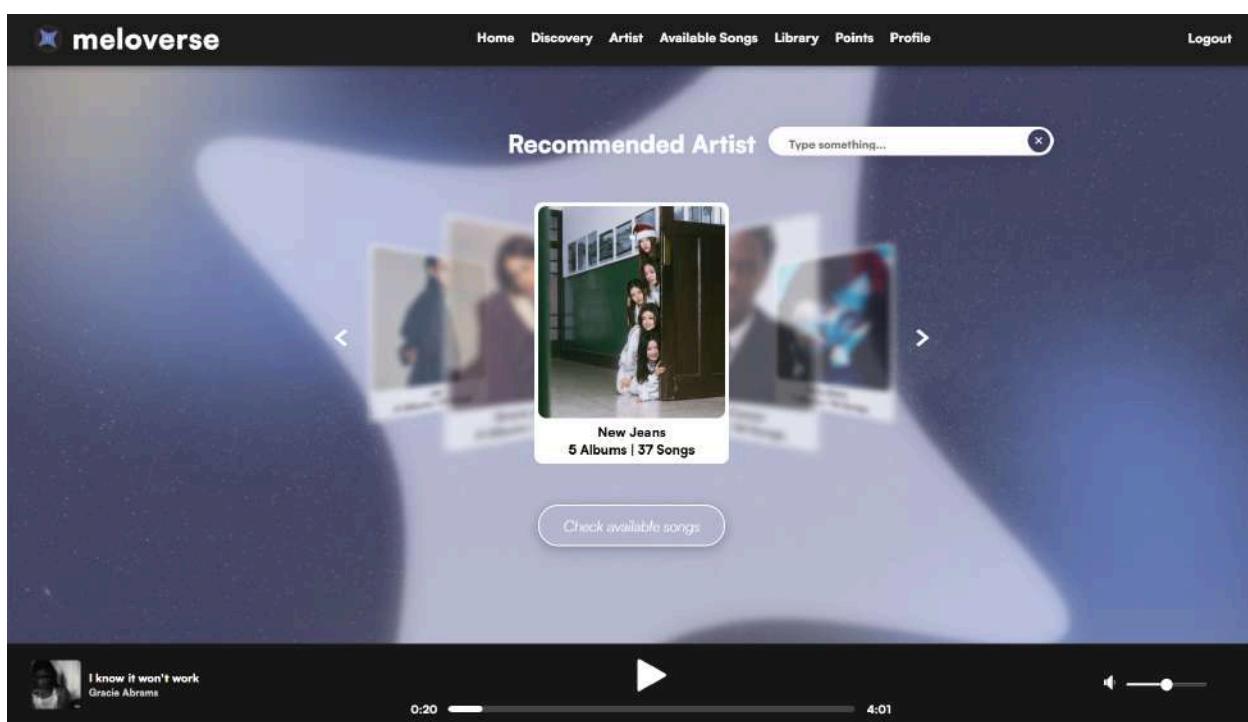
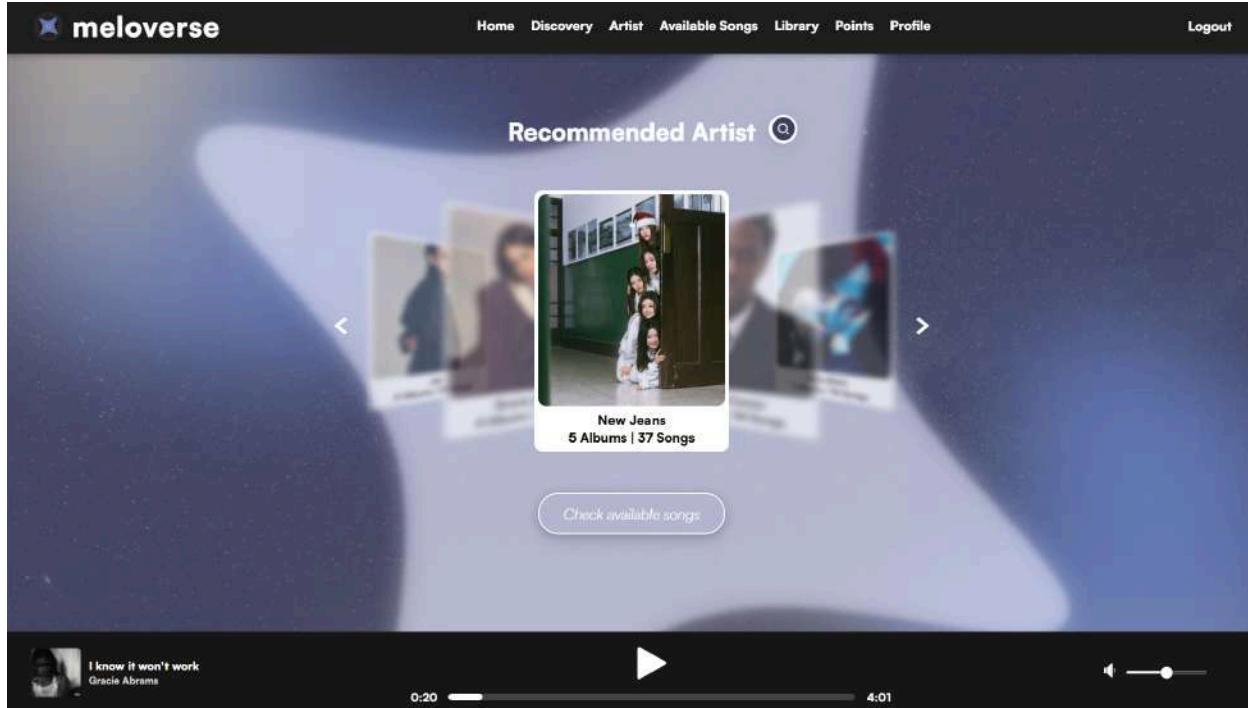
Rose

Happier Than Ever
Billie Eilish

0:32 4:58

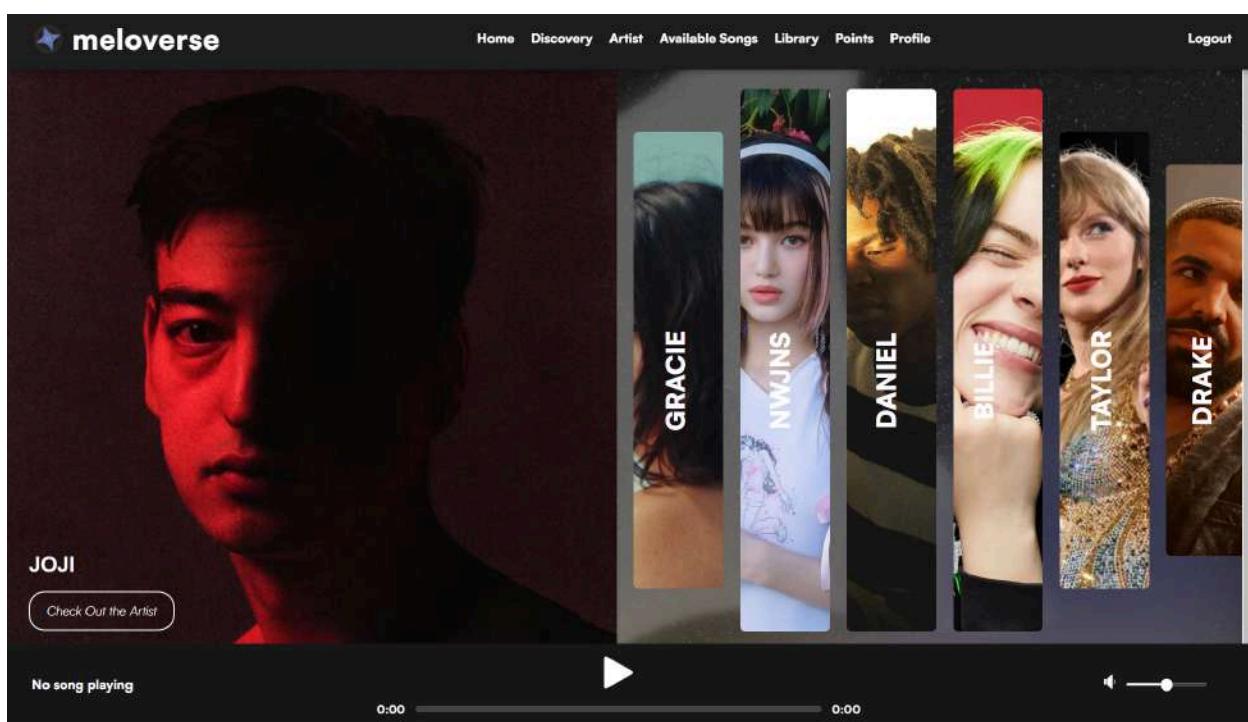
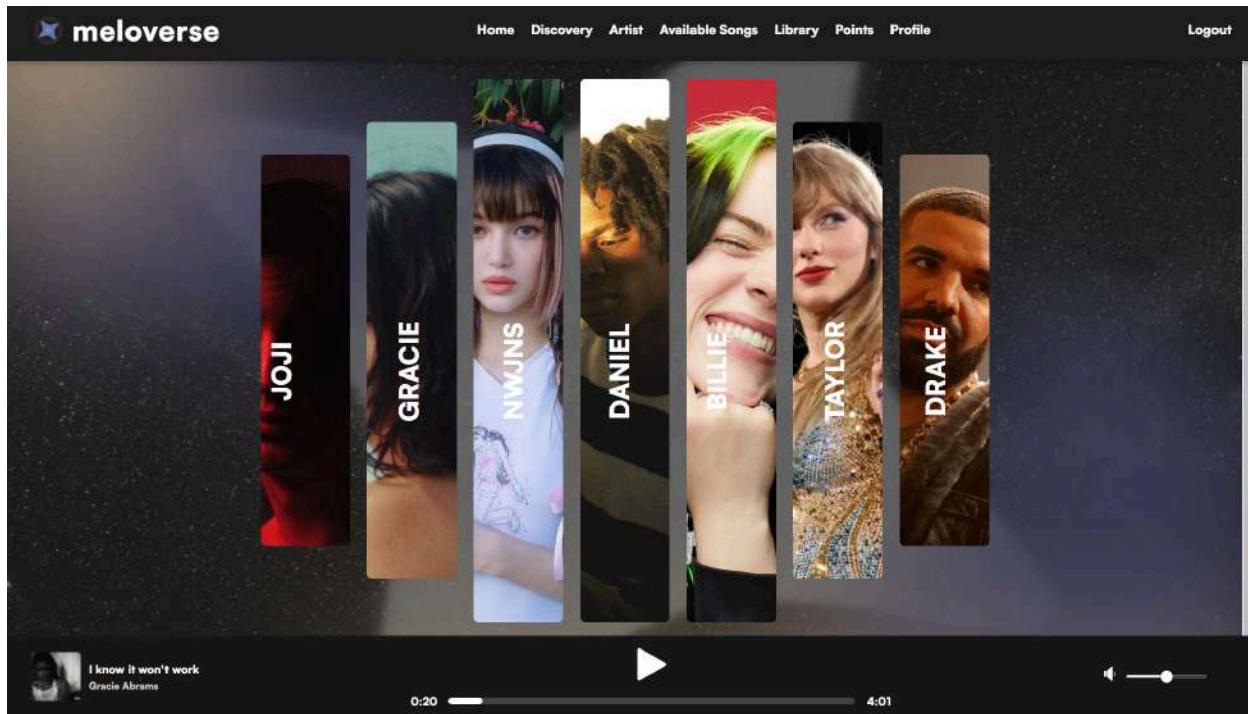
3.5 Discovery Page

The Discovery Page encourages exploration by showcasing new songs, albums, and artists. It is designed to help users discover trending music and emerging artists they may enjoy.



3.6 Artist Page

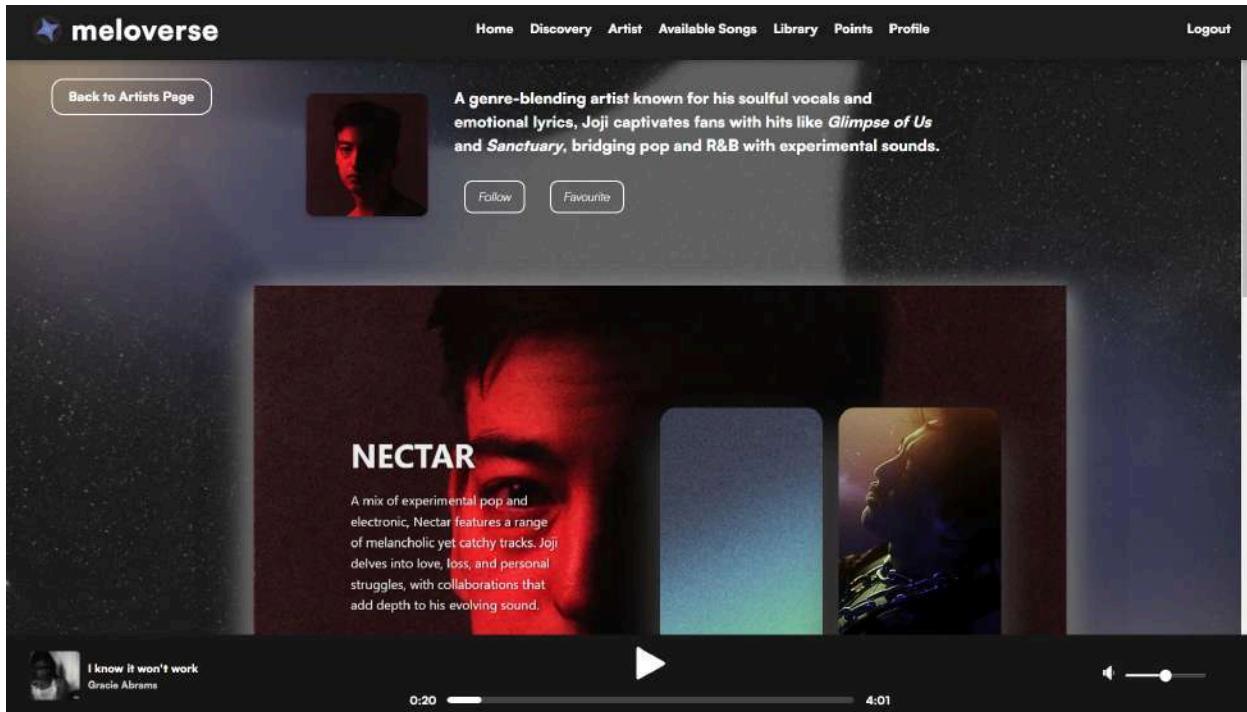
The Artist Page is a central feature of Meloverse, designed to showcase artists in a visually appealing and user-friendly manner.



It provides users with a comprehensive overview of their favorite musicians while encouraging deeper connections through content like albums, tracks, and merchandise. The page is carefully segmented into the following sections to ensure seamless navigation:

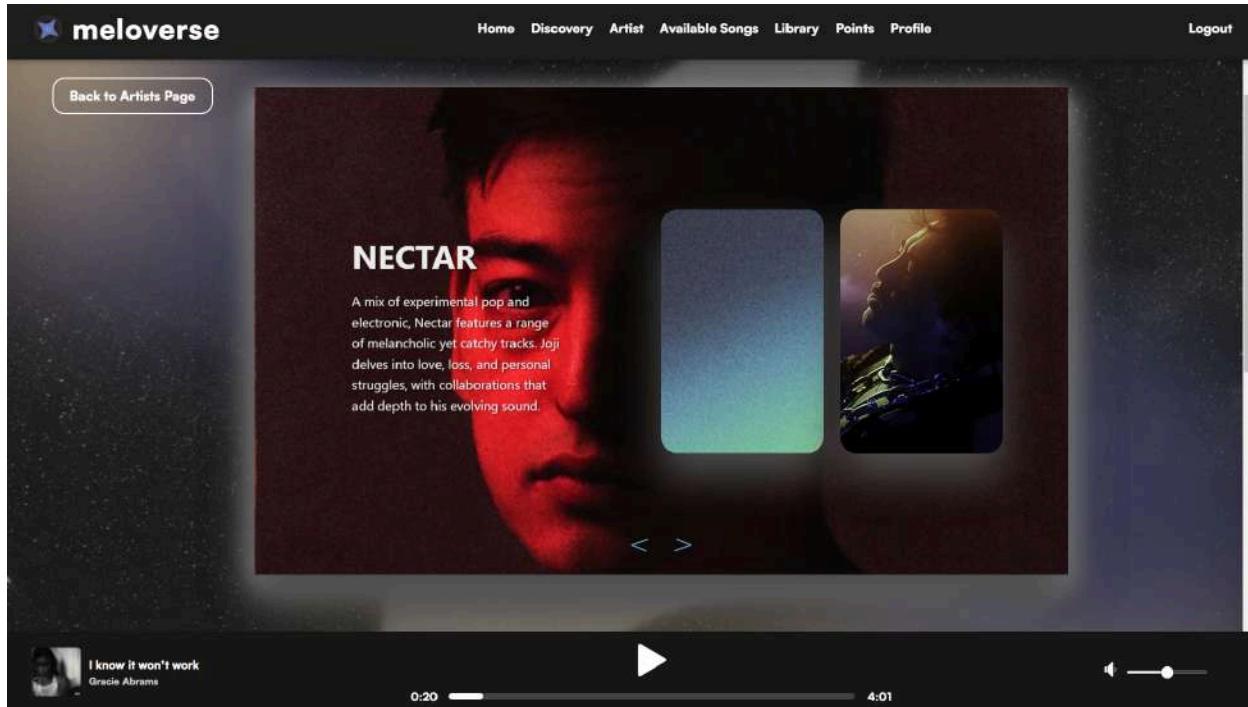
3.6.1 Artist Info

The Artist Info section provides an overview of the artist, including their biography, career highlights, and key achievements, offering fans insights into their musical journey.



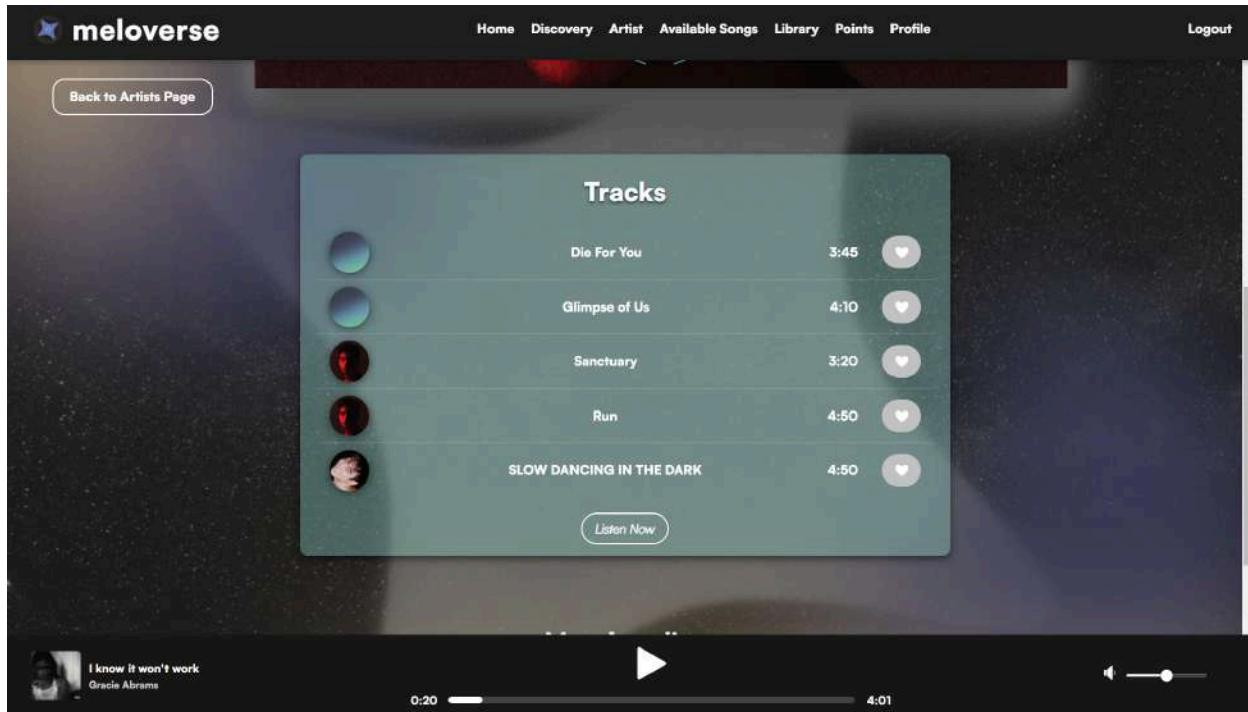
3.6.2 Artist Albums

The Artist Albums section displays the artist's discography with album covers and the information, allowing users to browse and explore their music effortlessly.



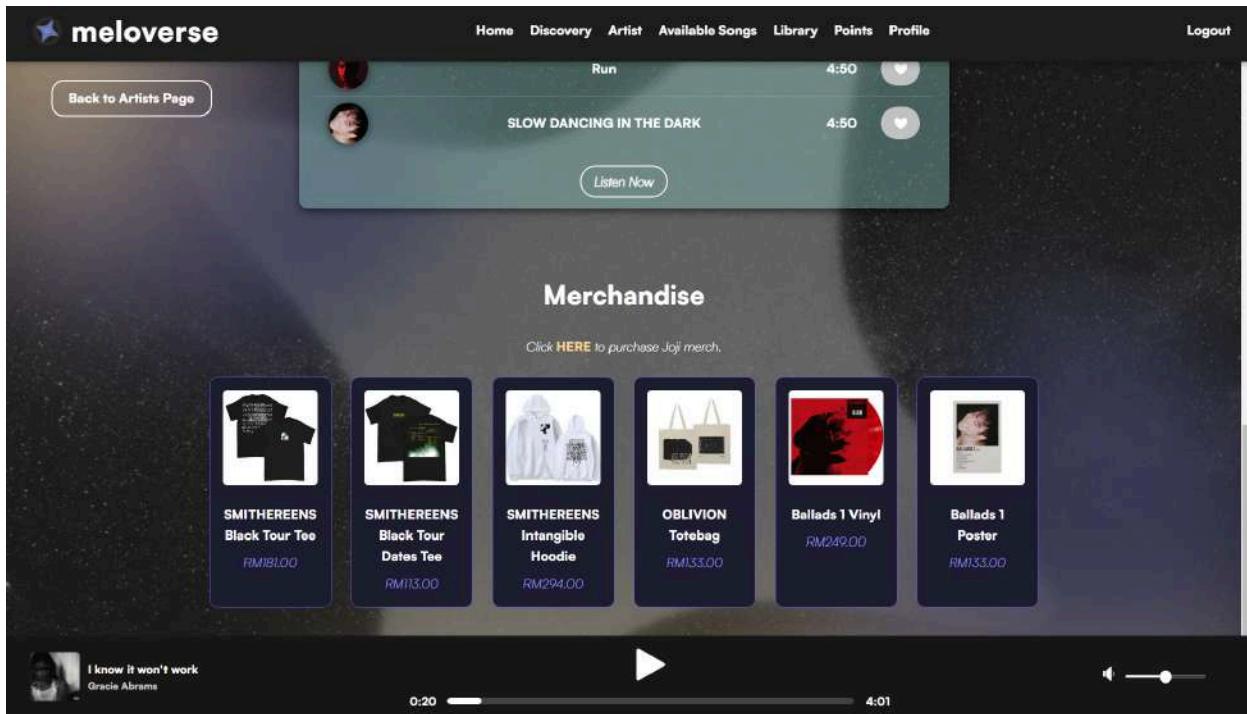
3.6.3 Artist Tracks

The Artist Tracks section highlights popular songs and allows users to play, pause, and listen to the artist's top tracks directly on the page.



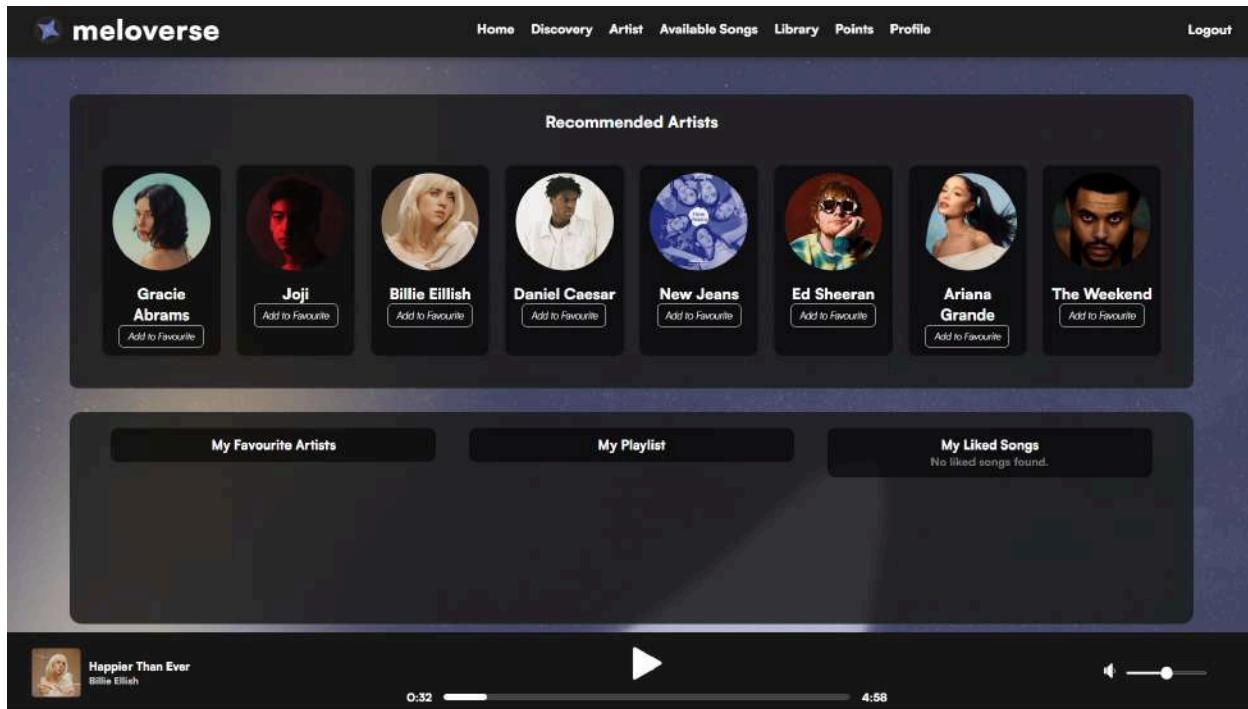
3.6.4 Artist Merchandise

The Artist Merchandise section showcases exclusive artist-branded products like clothing, accessories, and posters, providing fans an opportunity to purchase and support their favorite artist.



3.7 Library Page

The Library Page is a personalized section where users can organize and view their favorite songs, albums, and artists. It allows easy management of their musical collections.



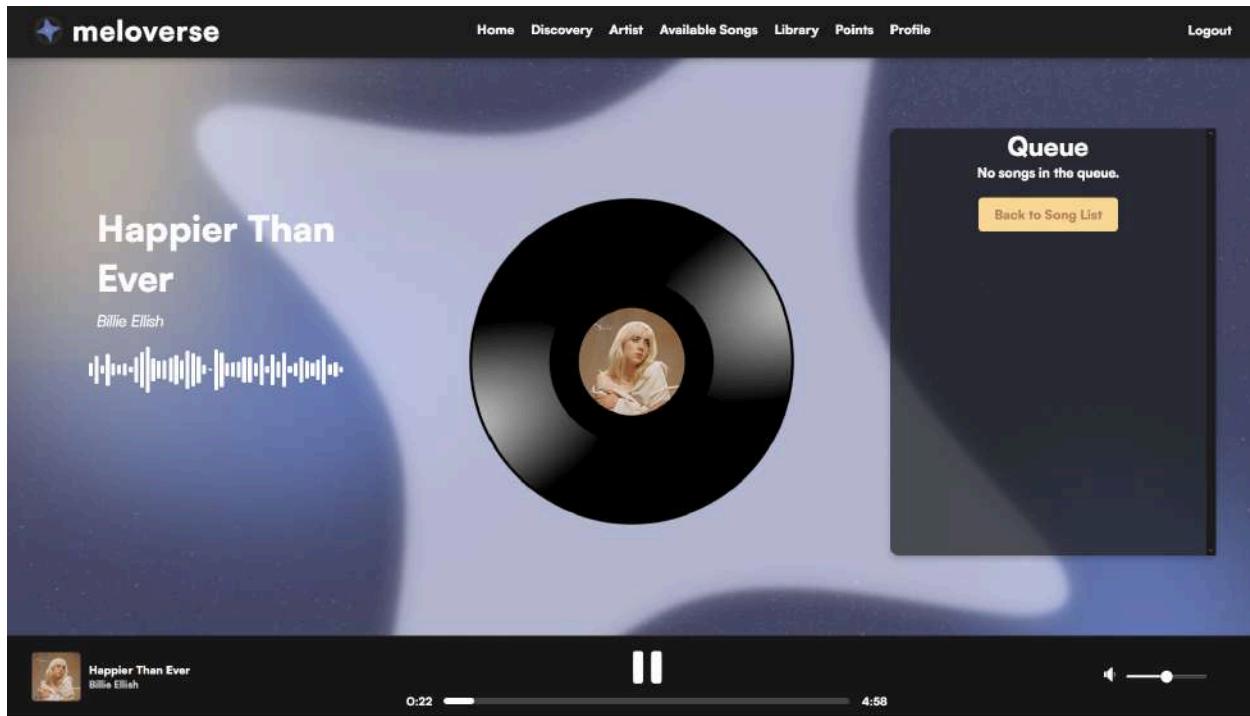
3.8 Song List Page

The Song List Page displays a list of available songs, enabling users to browse, search, and play their desired music seamlessly.

The screenshot shows the meloverse interface for the artist Billie Eilish. At the top, there's a navigation bar with links for Home, Discovery, Artist, Available Songs, Library, Points, and Profile, along with a Logout button. Below the navigation, the artist's name "BILLIE ELLISH" is displayed in bold capital letters. The page is organized into three main sections: "Happier Than Ever", "HIT ME HARD AND SOFT", and "WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?". Each section contains album covers, song titles, and small icons for adding to a library or sharing. In the bottom right corner, there's a media player showing a song by Gracie Abrams titled "I know it won't work". The player includes a play button, a progress bar from 0:20 to 4:01, and volume control.

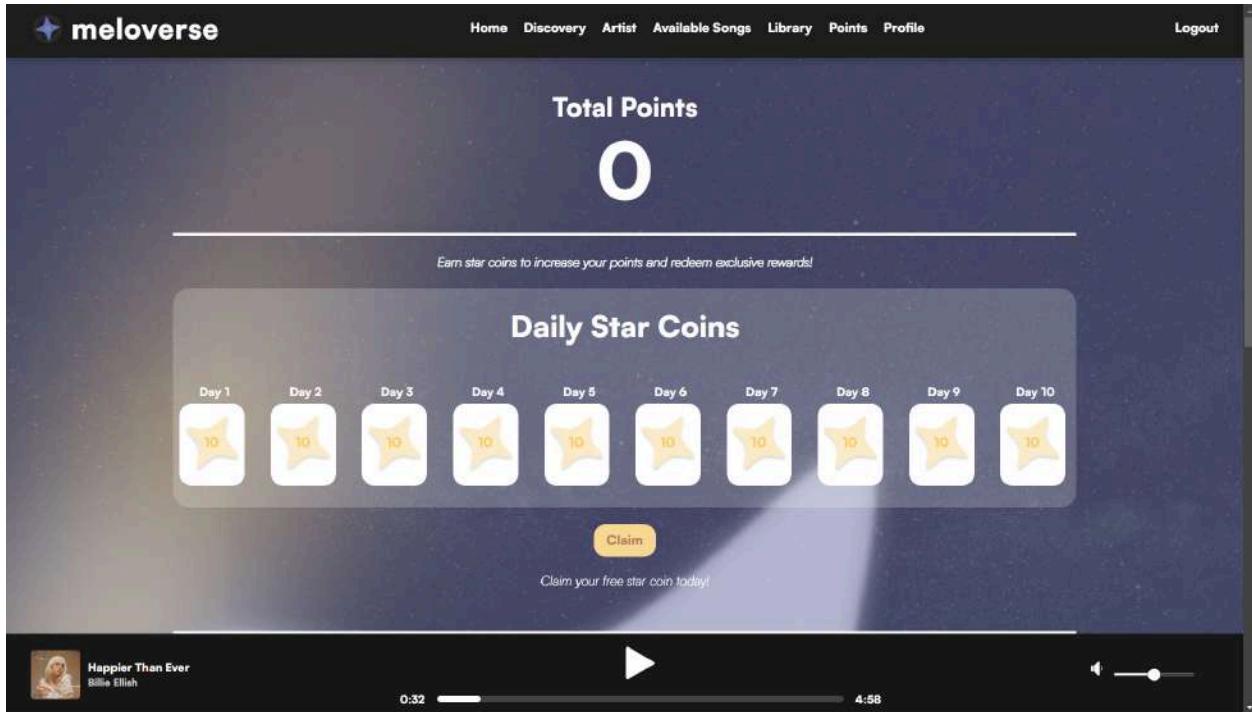
3.9 Song Page

The Song Page provides an immersive listening experience with song details, queue list, and user controls for managing their music.

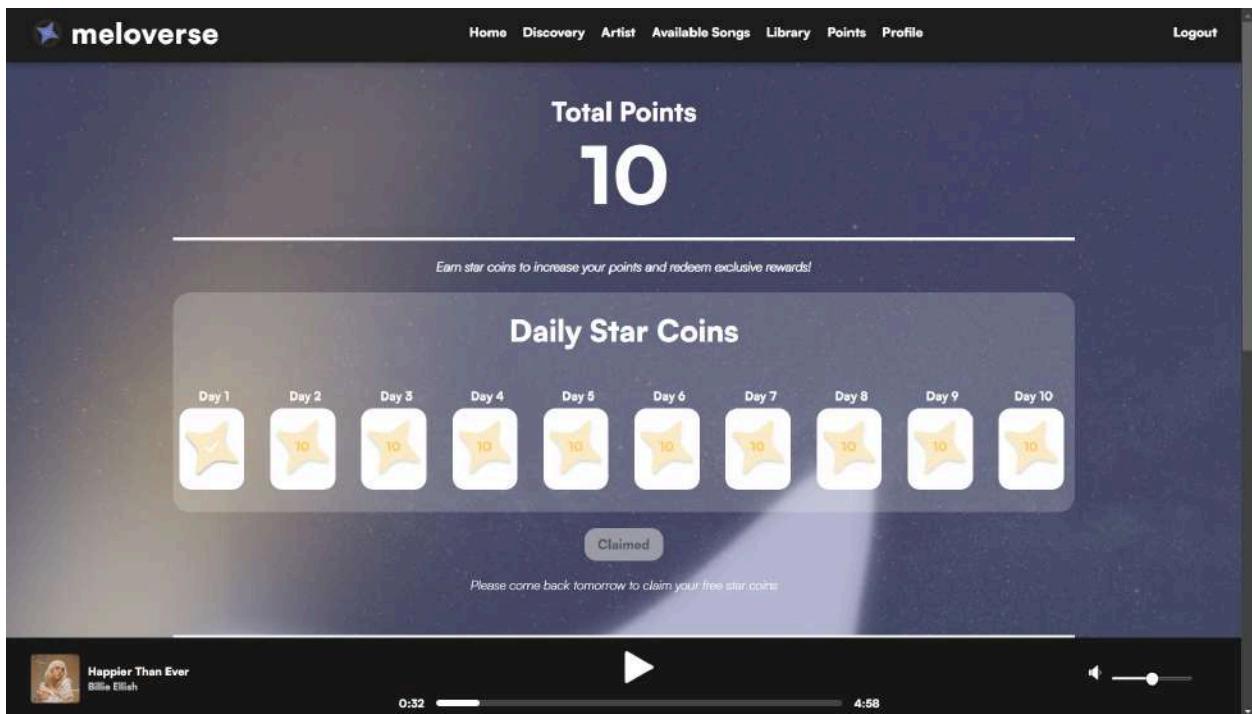


3.10 Points Page

The Points Page introduces an exciting rewards system where users can earn points through activities like login to Meloverse daily.



3.10.1 Points Claimed



3.10.2 Reward List

Points can be redeemed for various rewards, enhancing engagement and user satisfaction.

The screenshot shows the 'Reward List' screen within the meloverse mobile application. At the top, there is a navigation bar with the 'meloverse' logo, 'Home', 'Discovery', 'Artist', 'Available Songs', 'Library', 'Points', 'Profile', and 'Logout' options. Below the navigation bar, the title 'Reward List' is centered. The main content area displays eight reward cards arranged in two rows of four. Each card includes a reward description, required points, and a 'Redeem' button.

Reward Description	Required Points	Action
Free Music Download	100 points	Redeem
Free 1-Day Premium	200 points	Redeem
Exclusive Sticker Pack	300 points	Redeem
Access to Ad-Free Mode (3 Days)	400 points	Redeem
Exclusive Merchandise Discount	500 points	Redeem
Free Concert Ticket (Local Events)	750 points	Redeem
Gift Card for Popular Stores	1000 points	Redeem
Custom Playlist Created by Experts	1200 points	Redeem
Early Access to New Features	1400 points	Redeem

At the bottom of the screen, there is a dark navigation bar featuring a play button icon, a progress bar indicating '0:32' to '4:58', and a volume control icon.

3.11 Profile Page

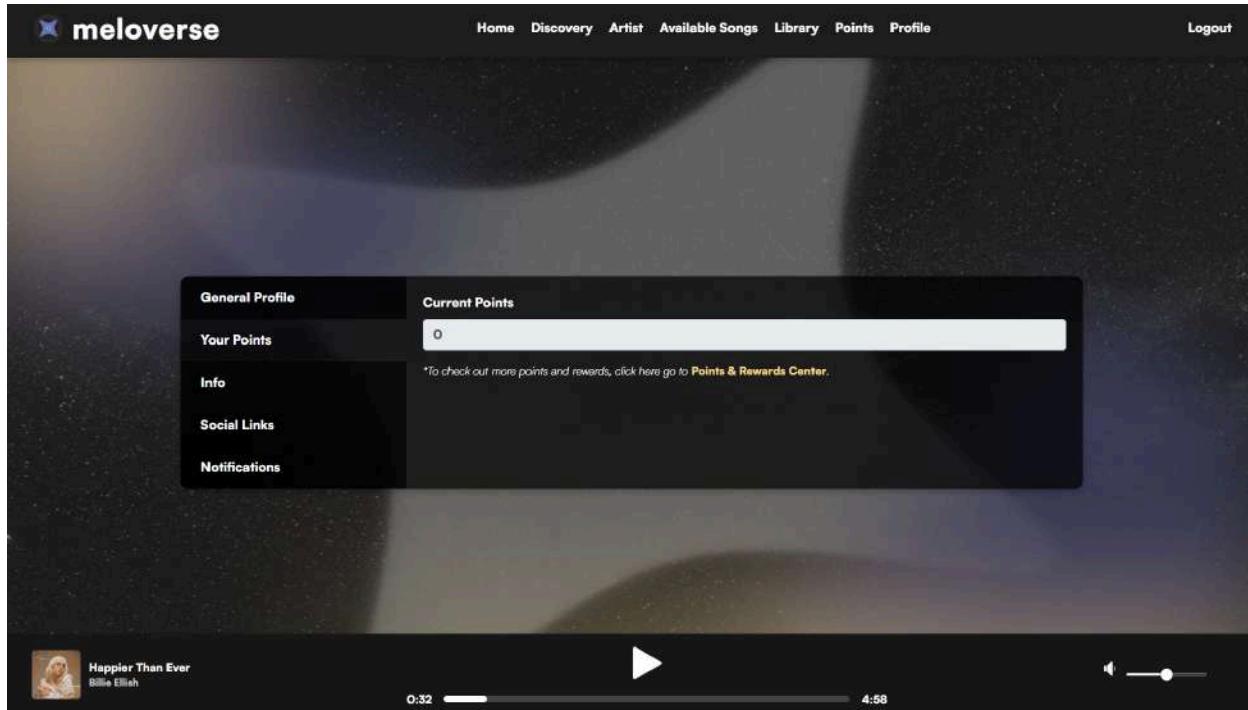
3.11.1 General Profile

The General Profile section serves as the main hub for user details, displaying essential information such as the user's name, email, and profile picture for a personalized experience.

The screenshot shows the 'General Profile' section of the meloverse application. At the top, there is a navigation bar with links for Home, Discovery, Artist, Available Songs, Library, Points, Profile, and Logout. On the left, a sidebar lists 'General Profile', 'Your Points', 'Info', 'Social Links', and 'Notifications'. The main area contains fields for 'First Name' (user1), 'Last Name' (testing), 'E-mail' (user1@gmail.com), and 'Phone Number' (0108323717). A message box indicates that the email is not confirmed and provides a 'Resend confirmation' link. Below these fields is a large 'Update Profile' button. At the bottom of the screen, there is a player interface showing a song by Billie Eilish titled 'Happier Than Ever' at 0:32 of a 4:58 track duration.

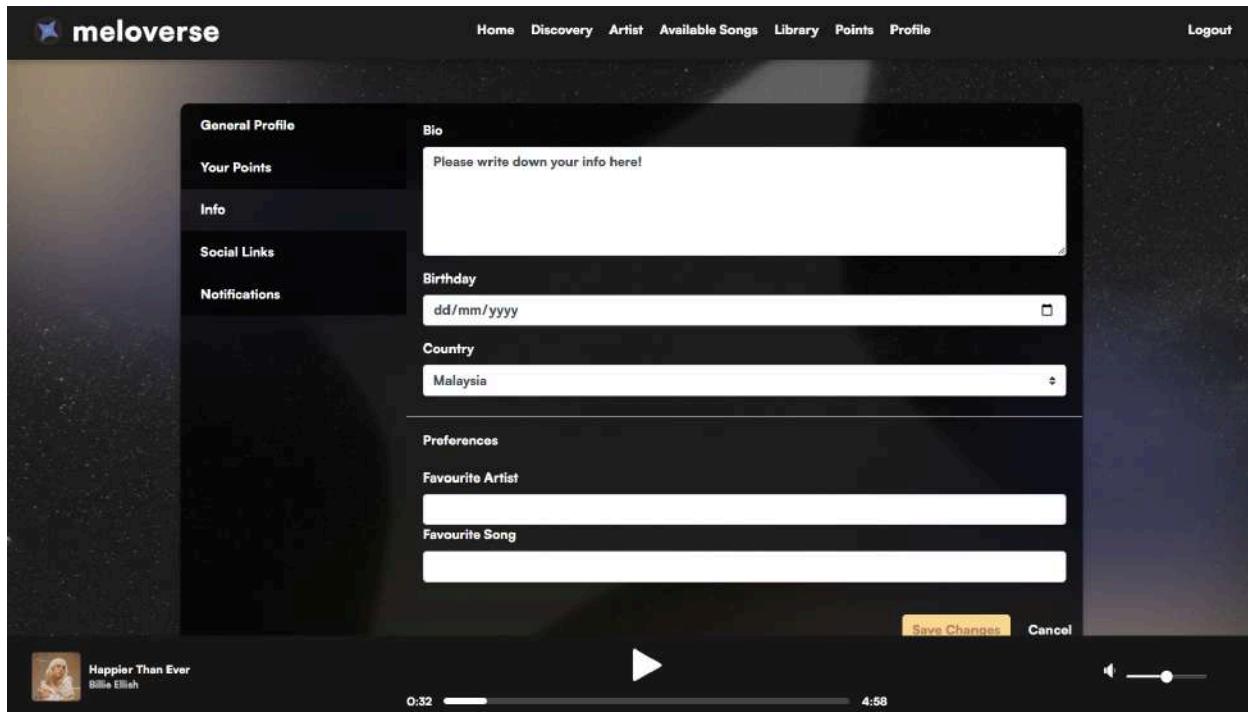
3.11.2 Your Points

The Your Points section allows users to track their reward points, offering a clear view of their progress toward redeeming exclusive rewards or incentives on the platform.



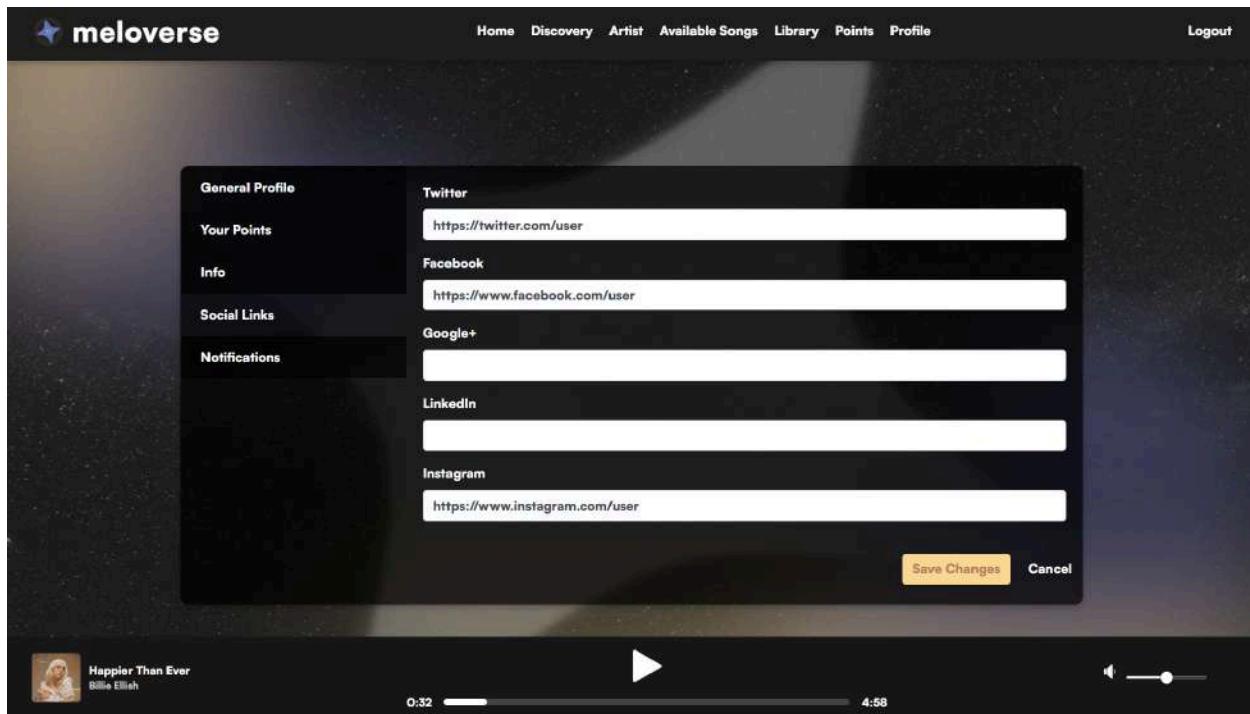
3.11.3 Info

The Info section provides users with additional details about their account and activity, ensuring they have all the necessary information in one accessible place.



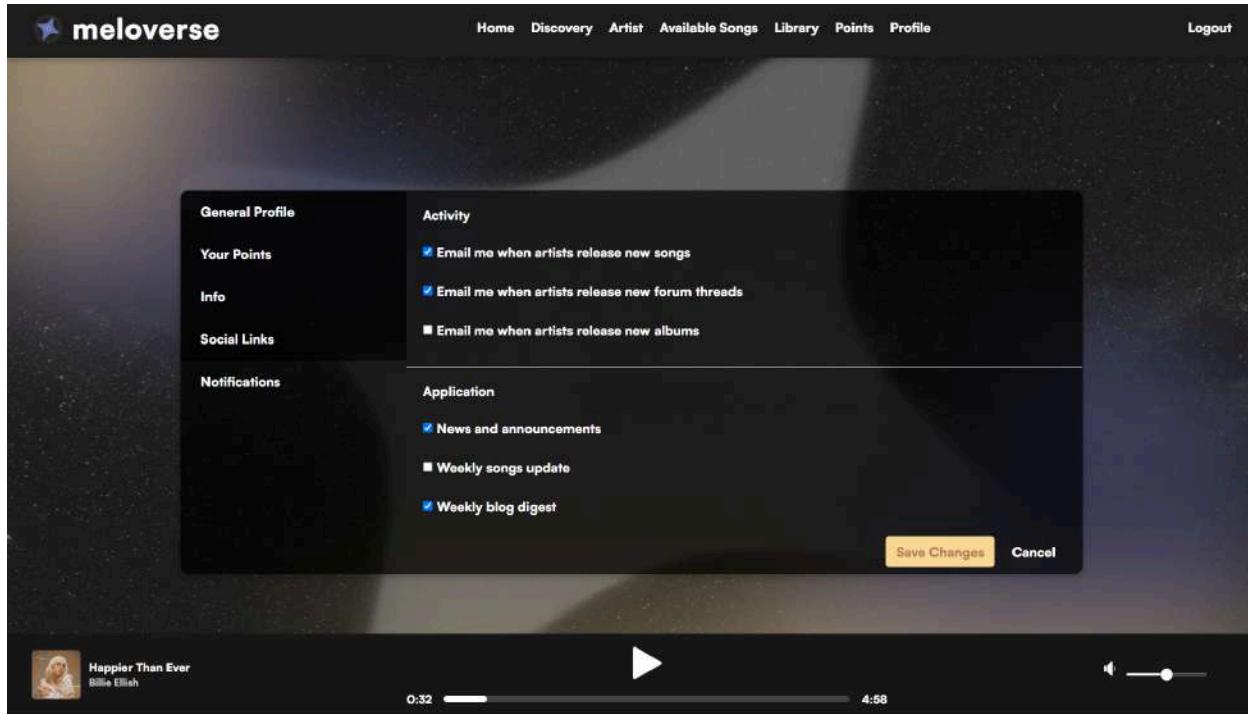
3.11.4 Social Links

The Social Links section lets users connect their social media accounts, enabling seamless sharing of music and engagement with their friends or community.



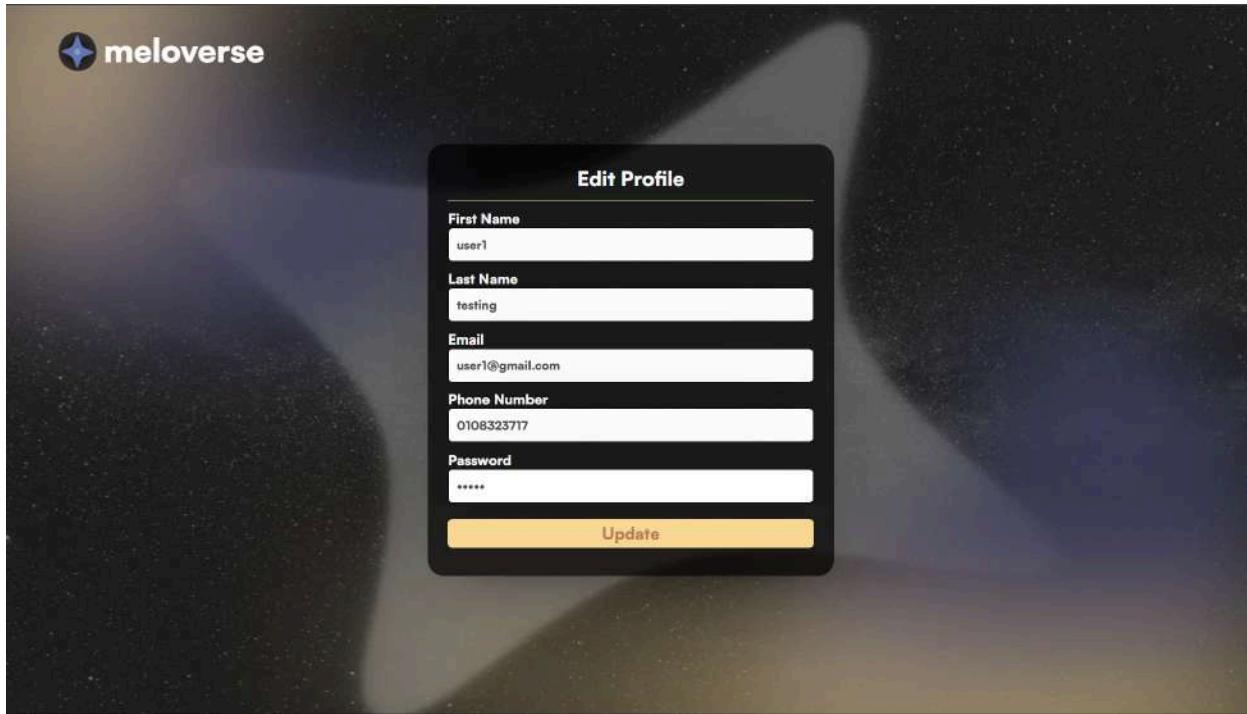
3.11.5 Notifications

The Notifications section keeps users updated with real-time alerts about new music releases, rewards, artist updates, and other platform activities.



3.12 Update Profile Page

The Update Profile Page allows users to modify their profile information, such as updating their name, photo, or preferences, ensuring their account stays current and personalized.



4.0 Functions and Features

4.1 SQL Database

Our database, named login, contains seven tables: artists, favourites, likedsongs, playlist, queue, songs, and users, designed to manage music-related data such as user details, songs, playlists, and favorite tracks. It uses the InnoDB storage engine with a utf8mb4_general_ci collation for efficient data storage and retrieval.

The screenshot shows the phpMyAdmin interface for the 'login' database. The left sidebar lists databases and tables:

- information_schema
- New
- login

 - artists
 - favourites
 - likedsongs
 - playlist
 - queue
 - songs
 - users

- midterm2
- mysql
- performance_schema
- phpmyadmin
- song
- test

The main area displays the table structure for the 'login' database:

Table	Action	Rows	Type	Collation	Size	Overhead
artists	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	16.0 KIB	-
favourites	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KIB	-
likedsongs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
playlist	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
queue	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
songs	Browse Structure Search Insert Empty Drop	25	InnoDB	utf8mb4_general_ci	32.0 KIB	-
users	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_general_ci	32.0 KIB	-

Below the table list, there is a 'Create new table' form:

Create new table

Table name: Number of columns: Create

Figure 1. Database

4.2 HTML5 form

Figure 2 shows us a HTML5 form snippet that creates a login interface for Meloverse, featuring input fields for email and password, with client-side validation and autocomplete disabled for enhanced security. It also includes a call-to-action link for users to navigate to the signup page if they do not have an account.

```
<!-- HTML Form -->
<header>Login to Meloverse</header>
<form action="" method="post">
    <div class="field input">
        <label for="email">Email</label>
        <input type="text" name="email" id="email" autocomplete="off" required>
    </div>

    <div class="field input">
        <label for="password">Password</label>
        <input type="password" name="password" id="password" autocomplete="off" required>
    </div>

    <div class="field">
        <input type="submit" class="btn" name="submit" value="Login" required>
    </div>
    <div class="links">
        Don't have account? <a href="signinPage.php">Sign Up Now</a>
    </div>
</form>
```

Figure 2. Code Snippet of LoginPage.php

4.3 SQL Commands

4.3.1 CREATE (INSERT)

```
if (mysql_num_rows(result: $verify_query) != 0) {
    header(header: "Location: emailError.php");
    exit();
} else {
    mysqli_query(mysql: $conn, query: "INSERT INTO users(fName,lName,Email,PhoneNo,Password) VALUES('$fname','$lname','$email','$phoneno','$password')") or die("Error Occured");
    header(header: "Location: register.php");
    exit();
}
else {
```

Figure 3. Code Snippet of signinPage.php

4.3.2 READ (SELECT)

```
$query = mysqli_query(mysql: $conn, query: "SELECT*FROM users WHERE Id=$id");

while ($result = mysqli_fetch_assoc(result: $query)) {
    $res_Fname = $result['fName'];
    $res_Lname = $result['lName'];
    $res_Email = $result['Email'];
    $res_PhoneNo = $result['PhoneNo'];
    $res_id = $result['Id'];
}

?>
```

Figure 4. Code Snippet of profilePage.php

4.3.3 UPDATE (MODIFY)

```
$edit_query = mysqli_query(mysql: $conn, query: "UPDATE users SET fName='$fname', lName='$lname', Email='$email', PhoneNo='$phoneno', Password='$password' WHERE Id=$id") or die("error occurred");

if ($edit_query) {
    echo "<div class='message'>
        <p>Profile Updated!</p>
    </div><br>";
    echo "<a href='profilePage.php'><button class='btn'>Go Back To Profile Page</button></a>";
}
else {
```

Figure 5. Code Snippet of updateProfile.php

4.3.4 DELETE

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['remove_id'])) {
    $remove_id = intval(value: $_POST['remove_id']);
    $delete_sql = "DELETE FROM queue WHERE id = $remove_id";
    if ($conn->query(query: $delete_sql)) {
        echo json_encode(value: ['status' => 'success']);
    } else {
        echo json_encode(value: ['status' => 'error', 'message' => $conn->error]);
    }
    exit();
}
```

Figure 6. Code Snippet of queue_display.php

4.4 Usage of sessions

In this code, PHP sessions are used to store and manage temporary user-specific data across multiple pages during the user's session. The `session_start()` function initializes the session and allows data to persist throughout the user's interaction with the web application.

Specifically, the variable `$_SESSION['current_song']` is used to keep track of the song that the user is currently interacting with or listening to. This ensures the state of the current song is preserved, enabling a seamless user experience. If the session data is unavailable, it defaults to null. On the other hand, the variable `$_SESSION['is_playing']` is used to store whether the song is currently playing, ensuring the application remembers the playback state. If this data does not exist in the session, it defaults to false.

Using sessions avoids the need to pass such data between pages using URLs or form submissions, thereby enhancing security and usability. Sessions are essential for maintaining stateful data in a stateless protocol like HTTP, ensuring a more personalized and dynamic experience for the user.

```
<?php
// Database connection
session_start();
// Assuming user authentication has been handled
$host = "localhost:3307";
$user = "root";
$pass = "";
$db = "login";
$conn = new mysqli(hostname: $host, username: $user, password: $pass, database: $db);

if ($conn->connect_error) {
    die("Failed to connect DB: " . $conn->connect_error);
}

$current_song = isset($_SESSION['current_song']) ? $_SESSION['current_song'] : null;
$is_playing = isset($_SESSION['is_playing']) ? $_SESSION['is_playing'] : false;
$conn->close();

?>
```

Figure 7. Code Snippet of `homePage.php`

4.5 Protection against SQL injection

```
$stmt = $conn->prepare(query: "SELECT * FROM users WHERE Email = ? AND Password = ?");  
$stmt->bind_param(types: "ss", var: &$email, vars: &$password);
```

Figure 8. Code Snippet of loginPage.php

The code utilizes prepared statements and parameterized queries to protect against SQL injection attacks. By using the `prepare()` and `bind_param()` methods, the user inputs (e.g., email and password) are treated as parameters rather than executable SQL code. This ensures that any malicious input, such as "`" OR 1=1 --`", is not executed by the database engine but instead treated as a literal string.

Prepared statements safeguard the application by separating SQL logic from user data, thereby preventing attackers from injecting malicious SQL queries. This approach adheres to best practices in secure database interactions and significantly reduces the risk of unauthorized access or database manipulation.

4.6 Protection against HTML injection

```
$_SESSION['valid'] = htmlspecialchars(string: $row['Email']);  
$_SESSION['fname'] = htmlspecialchars(string: $row['fName']);  
$_SESSION['lname'] = htmlspecialchars(string: $row['lName']);  
$_SESSION['phoneno'] = htmlspecialchars(string: $row['PhoneNo']);  
$_SESSION['id'] = htmlspecialchars(string: $row['Id']);
```

Figure 9. Code Snippet of loginPage.php

The application uses the `htmlspecialchars()` function to sanitize output data fetched from the database before storing it in session variables. This encoding converts special characters, such as `<`, `>`, and `"`, into their HTML entity equivalents (e.g., `<`, `>`, `"`).

By sanitizing user-generated content, the application prevents malicious scripts from being executed when the data is displayed in the browser. This measure effectively mitigates Cross-Site Scripting (XSS) attacks, ensuring that any injected code is treated as plain text rather than executable HTML or JavaScript. This approach is critical for maintaining the integrity of the user interface and protecting end users from potential exploits.

4.7 CSS Usage

CSS (Cascading Style Sheets) is used in this project to enhance the visual presentation and user experience of the web application. By separating the content (HTML) from the design (CSS), the code achieves better maintainability and readability.

In the provided code down below:

```
<link rel="stylesheet" href="/Web Application Programming/CSS/loginPage.css">
```

Figure 10. Code Snippet of loginPage.php

For example, the external CSS file loginPage.css is linked using the `<link>` tag to define the styles for the login page elements such as the form, input fields, buttons, and overall layout. This ensures consistent styling across the page. Another CSS file, loginCarousel.css, is used to style the carousel feature that displays images dynamically. This adds interactivity and an aesthetically pleasing design for the users.

Custom styling improves the user interface (UI) by organizing content, defining colors, fonts, spacing, and responsiveness, thereby making the web application visually appealing and user-friendly. Using external CSS allows for reusability and modularity, where the same styles can be applied to multiple pages. It also promotes faster page loading since the CSS file is cached by the browser. Provided below Figure 11 displays all of the CSS we used.

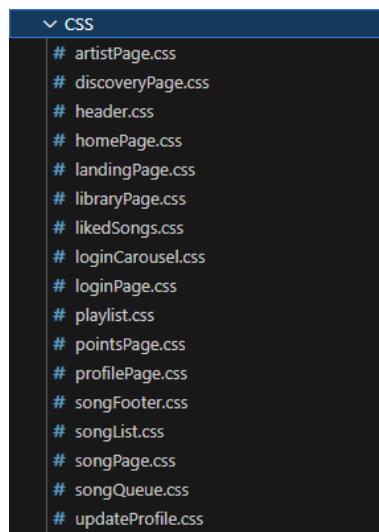


Figure 11. CSS list

4.8 AJAX

4.8.1 AJAX Call for Song Removal:

When the "Remove" button is clicked, an AJAX POST request is sent to the same script (queue_display.php) with the remove_id of the song to be removed from the queue table. The server-side script processes the request, deletes the song from the queue, and sends a JSON response indicating success or failure.

```
$.ajax({
    url: '/Web Application Programming/php/queue_display.php',
    type: 'POST',
    data: { remove_id: queueId },
    success: function(response) {
        console.log(response); // Debug server response
        const result = JSON.parse(response);
        if (result.status === 'success') {
            parentItem.remove();
            if ($('#queue-list .queue-item').length === 0) {
                $('#queue-list').html('<p>No songs in the queue.</p>');
            } else {
                alert('Error removing song: ' + result.message);
            }
        },
        error: function() {
            alert('An error occurred. Please try again.');
        }
});
```

Figure 12. Code Snippet of queue_display.php

4.8.2 Server-side Handling:

The PHP script checks if the request method is POST and if remove_id is set. If so, it performs a DELETE query on the queue table to remove the specified entry.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['remove_id'])) {
    $remove_id = intval(value: $_POST['remove_id']);
    $delete_sql = "DELETE FROM queue WHERE id = $remove_id";
    if ($conn->query(query: $delete_sql)) {
        echo json_encode(value: ['status' => 'success']);
    } else {
        echo json_encode(value: ['status' => 'error', 'message' => $conn->error]);
    }
    exit;
}
```

Figure 13. Code Snippet of queue_display.php

4.9 JSON

4.9.1 Server-to-Client Communication

When a song is removed from the queue, the PHP server script sends a response in **JSON format**. This happens in the following code snippet:

```
if ($conn->query(query: $delete_sql)) {
    echo json_encode(value: ['status' => 'success']);
} else {
    echo json_encode(value: ['status' => 'error', 'message' => $conn->error]);
}
exit;
```

Figure 14. Code Snippet of queue_display.php

The PHP `json_encode()` function is used to create a JSON-encoded response. The response includes:

- status: Indicates whether the operation succeeded (success) or failed (error).
- message: If there's an error, the error message is included.

4.9.2 Client-side Handling:

The parsed JSON object (result) is used to determine the next actions:

- If `result.status === 'success'`, the song is removed from the UI.
- If `result.status === 'error'`, an alert displays the error message.

Example of handling JSON response in JavaScript:

```
success: function(response) {
    console.log(response); // Debug server response
    const result = JSON.parse(response);
    if (result.status === 'success') {
        parentItem.remove();
        if ($('#queue-list .queue-item').length === 0) {
            $('#queue-list').html('<p>No songs in the queue.</p>');
        }
    } else {
        alert('Error removing song: ' + result.message);
    }
},
```

Figure 15. Code Snippet of queue_display.php